INTRODUCTION TO FORMAL SEMANTICS

Email: wlechner@gs.uoag.gr (updated 2014)
Course webpage:http://users.uoa.gr/~wlechner/Semwien.htm
Table of Content: *see appendix*

---

1. INTRODUCTION[1]

Distribution of Labor:

- *Syntax* investigates the recursive part-whole structure of natural language expressions. More generally, syntax is concerned with properties of a formal system, its *form*.
- *Natural languages semantics* (NLS) studies the meaning of natural language (NL) expressions. In general, semantics focuses on relations of a formal system to its interpretation, its *content.*
- *Pragmatics* is concerned with the use of expressions in context, and is in this sense a term specific to linguistic. What are the parts of meaning of an utterance which may possibly change depending upon the situation?

1.1. THREE OBJECTIVES OF NLS

I. Representing meaning

What is meaning? What do sentences *denote* (= mean)? What is meaning, and how is it represented in the mind? Speakers e.g. understand (1), even though there is no such rule in the real world that would regulate any game.

(1) The three players alternate in making seven moves at a time until the game is completed.

What do words refer to? The English *existential construction*, exemplified by (2)a, asserts the existence of the DP in the *coda* position. Why is it then that the coda can be filled by something that does not exist, as in (2)b?

(2) a. There is [$_{\text{DP coda}}$ a smallest prime number] ≈
≈ A smallest prime number exists
b. There is [$_{\text{DP coda}}$ no largest prime number]

How are the meanings of the words related to the sentence meaning?

---

[1]These notes are work in progress and in part follow Heim and Kratzer (199*8*). Additional ideas are adopted from Partee et. al (1990), lecture notes by Arnim von Stechow and Ede Zimmerman (see references) and Sabine Iatridou (p.c.).

II. Excluding ill-formed strings (a side product of I):

Insights into the principles underlying natural language semantic (NLS) make it possible to account for the ill-formedness of the starred examples below:

● Exceptives:

(3) a. The Pope invited everyone - but not Sam
    b. The Pope invited everyone except Sam

(4) a. The Pope invited someone/a cardinal - but not Sam.
    b. *The Pope invited someone/a cardinal except Sam

(5) a. All natural numbers between 0 and 3 except 2 are odd.
    b. *A/some/no natural number between 0 and 3 except 2 is odd.

● Aspect and temporal/durative adverbs:

(6) a. They solved the equation in an hour
    b. *They solved the equation for hours
    c. *They solved equations in an hour
    d. They solved equations for hours

(7) a. They discussed the treatment of the illness in an hour
    b. *They discussed the treatment of the illness for an hour
    c. *They discussed the treatment of illnesses in an hour
    d. They discussed the treatment of illnesses for hours

● Quantifiers:

(8) a. Every second player wins
    b. Not every player wins
    c. *Not every second player wins (with regular intonation, i.e. no focus on *every* or *second*)

(9) a. Sam and Sally were invited to the party
    b. All participants and some guests were invited to the party
    c. No participants and few of the guests were invited to the party
    d. *No participants and Sam were invited to the party

III. Representing inferences

Whoever knows (10)a and (10)b also knows that (10)c is true. How are the meanings of the individual sentences related?

(10) a. The candidate must be over 25 years old and must be Russian
    b. Mary remembered that Sam was born in Stockholm
    c. Sam cannot be the candidate

● While (11)b follows from (11)a, (12)a does not entail (12)b. Why?

(11) a. The famous actor who stopped smoking was a radical Scientologist
    b. The famous actor used to smoke

(12) a. Everybody who stopped smoking was a radical Scientologist
    b. Everybody used to smoke

1.2. Motivating NLS

Fundamental objective of NL-semantics is to define the meaning of utterances such as (13) - (16).

(13) A representative$_k$ of every southern county$_i$ that has ever voted for the Democrats believed that he$_k$ has contributed to its$_{i/j}$ development. (let j = the new pharmaceutical)

(14) Nothing is good enough for you

(15) Exactly half of the squares contain less than three triangles

(16) Not every boy can be above average height

More precisely, NL Semantics investigates, among others, the following questions:

❐ How can the situations be best described in which a sentence can be truthfully uttered?
In which situations exactly is (13) intuitively thought to be true, for example?

❐ What are the procedures that make it possible to formalize the meaning of sentences?
What is the algorithm translating e.g. (13) into the quasi-mathematical form in (17), which is generally taken to resemble its (extensional first order) logic representation?

(17) $\forall$x[[x is a southern county $\wedge$ x voted for the Democrats] $\rightarrow$ $\exists$y[y is a representative $\wedge$ y believes that y contributed to x's development]]

❐ How does the meaning of a sentence relate to its syntactic form?

1.2.1. Syntax vs. Semantics

While it is a rather uncontroversial assumption that syntax and semantics interact (the latter interprets e.g. specific utterances with a specific syntactic structure), it is less obvious how the two systems contribute to the form and representation of sentences. As a preliminary observation, it can be shown that the strong view that syntax is entirely autonomous from semantics (a hypothesis entertained to a certain degree by Chomsky) is not likely to succeed. In particular, there are aspects of the meaning of a sentence which are conditioned by syntax, and v.v. This can be seen by minimal variations of (13) which lead to unacceptability.

○ First, observe that form has an impact on meaning. For instance, the fact that the second quantifier in (13) is contained in a PP, and not in a relative clause, as in (18), is relevant for obtaining the intended interpretation, according to which representatives may vary w.r.t. counties.

(18) *A representative$_k$ which represents *every southern county$_i$* that has ever voted for the Democrats believed that he$_k$ has contributed to *its$_i$* development.

○ Second, properties related to the meaning component can also have an impact on the (abstract) syntax of a sentence. In (19), the conditions on the particle *ever* somehow seem to be violated.

(19) *A representative$_k$ of *some* southern county$_i$ that has ☛ ever voted for the Democrats believed that he$_k$ has contributed to its$_{i/j}$ development.

More precisely, the distribution of these so-called Negative Polarity Items (NPIs) like *ever* cannot be captured in terms of syntactic properties of the expression. It is e.g. not possible to posit a single *syntactically* motivated feature for all the contexts which tolerate NPIs, as becomes obvious from the complex patterns in (20) and (21). (other NPI's include *budge an inch*, *any, care to*, *can stand,...*):

(20) a. Sam hasn't ever heard about it
b. Nobody has ever heard about it.
c. Everybody who has ever heard about it likes it.
d. At most two dozen people have ever heard about it.

(21) a. *Sam has ever heard about it.
b. *Everybody has ever heard about it.
c. *Somebody who has ever heard about it likes it.
d. *At least two dozen people have ever heard about it.

This indicates that NPI licensing is subject to semantic, and not syntactic, conditions.

Moreover, semantic properties can also directly influence the (abstract) syntactic representation of an expression. In (22), the presence of a negative NP in some way seems to block the link between *every southern county* and the pronoun *its*. Since this link is generally thought to be mediated by an abstract representation altering c-command relations (LF), (22) can be taken to support the view that semantic properties also contribute to the shape of syntactic representations.

(22) **No* representative$_k$ of every southern county$_i$ believed that he$_k$ has contributed to ☛ its$_i$ development.

Crucially, in both cases above the conditions responsible for unacceptability cannot be found in syntactic properties of the sentences, but are semantic in nature: they refer to properties of the meanings of the expression (*every* vs. *some* in (13) vs. (19) and *every* vs. *no* in (13) vs. (22)). Thus, these phenomena fall into the domain of study of a field of its own, i.e. NL-semantics.

1.2.2. Some further cases demanding semantic analyses

● Distribution of prenominal attributive adjectives and predicative adjectives does not follow from syntactic restrictions:

(23) a. the young murderer
b. The murderer is young.

(24) a. the alleged murderer
b. *The murderer is alleged.

● Cross-linguistic semantic variation: why, e.g., can the English utterance in (25) be understood as in (27)a, while the German equivalent (26)a must be interpreted as in (27)b? (All the examples should be read with neutral intonation, i.e. without assigning negation a pitch accent).

(25) All that glitters isn't gold (Lasnik 1972)

(26) a. Sie meinte, daß alles was glänzt nicht Gold ist
b. Sie meinte, daß nicht alles was glänzt Gold ist

(27) a. ‘Not all things that glitter are things that consist of gold’
➜ correctly accounts for the fact that brass instruments glitter, but are not made of gold

b. ‘All things that glitter are things that do not consist of gold’
➜ incorrectly claims that good flutes (which are all made of gold) are not made of gold

● Classical Aristotelian and scholastic logic is mainly concerned with the shape of valid inferences.

(28) John made a sandwich on Friday in his office (rule of *Simplification*)
∴ John made a sandwich on Friday
∴ John made a sandwich

NOTATIONAL CONVENTION: read ∴ as: ‘it follows from the above that...’

(29) Sam is younger than Jeff
Jeff is younger than Bart
∴ Sam is younger than Bart

(30) No bird has a steering wheel (*Syllogism - more specifically,*
Every duck is a bird *the one called ‘Celarent’; see below*)
∴ No duck has a steering wheel

Classical logic fails to provide solutions for many phenomena, including quantification with *most* ((31)), sentences with multipe quantifiers ((32)), or the behavior of so called *donkey pronouns* as in (30).

(31) Most dogs sleep

(32) Noone liked every movie

(33) If a man is in Megara, he is not in Athens (Chrysippus, school of Megara, 6ct. BC)

To illustrate, Aristotelian logic cannot account for the meaning of quantifiers headed by *most* as the do not lend themselves to an analysis in terms of the sentential connectives $\wedge$ and $\rightarrow$:

(34) a. Most x [dog(x) $\rightarrow$ sleep(x)]
“Most individuals are such that if they are dogs then they sleep”
➜Vacuously true in situations with more non-dogs than dogs

b. Most x [dog(x) $\wedge$ sleep(x)]
“Most individuals are such that they are dogs and sleep”
➜ Trivially false in situations with more non-dogs than dogs

2. NATURAL LANGUAGE VS. FORMAL LOGIC

(partially from Partee et al. 1990)

NL semantics differs from the semantics of formal systems such as predicate logic in substantial ways. These differences are important in two respects: first, they indicate that NL semantics cannot be reduced to formal logic, legitimating the study of semantics from a linguistic perspective. (This insight has actually motivated some founders of modern logic, most notably Alfred Tarski (Polish logician, 1902-83), to exclude NL phenomena from a formal analysis.) Second, systematic differences between formal and NL semantics can aid in gaining a better understanding of properties of the linguistic system.

2.1. ILLOCUTION

Logic consists only of declaratives. Natural language also employs imperatives, questions and the like. It is not the case that the two systems are incompatible, though. Questions in natural language can be reconstructed as sets of propositions in second-order predicate logic:

(35) a. Which dog sleeps ?
b. $\{p|\ \exists x\ [[\![dog]\!]\ (x)(w) \wedge p = [\![sleep]\!]\ (x)]\}$

2.2. EXPLICITNESS

Natural language does not need to realize all interpreted expression overtly, it licenses ellipsis (e.g. Gapping, Stripping, VP-ellipsis and the like; see Fox 2000; Oirsouw 1987; Pesetsky 1982; Sag 1976):

(36)	a.	Gapping:	John likes beans, and Bill ⌂ rice	(⌂ = likes)
	b.	Stripping:	They met somebody, but they won't tell who ⌂	(⌂ = they met)
	c.	VP-Ellipsis:	They like rice, and we do ⌂, too	(⌂ = like rice)

2.3. (LOGICAL) CONNECTIVES

Formal languages (Propositional Calculus, Predicate Calculus, ...) only employ logical connectives (*and, or, not, if...then*), but not such particles as *because, while, since, after, whenever*, which carry additional information apart from their logical contribution. For instance, in one of its uses, the connective *but* serves as logical conjunction ($\wedge$), but marks what might be called a contrast of expectations. That is, when two sentences are joined by *but*, as in *The food was fresh but the bread was old*, the speaker wants to convey the information that the content of the first conjunct creates expectations with are not met by the content of the second one. This additional component of meaning can be captured by a model which incorporates several different layers of meaning (such as pragmatics along 'logical' semantics).

● In addition, connectives in NL do not always have the same meaning as logical connectives. For instance, the meaning of *and* seems to include a condition on temporal ordering of the conjuncts which is absent in the classical definition of *and* (in propositional calculus).

(37) a. She attracted hepatitis and spent a week in the hospital.
➜ the outbreak of the illness *precedes* the stay at the hospital

b. She spent a week in the hospital and attracted hepatitis.
➜ the outbreak of the illness *follows* the stay at the hospital

1.3. AMBIGUITY

Expressions in natural language can be ambiguous, while formal languages are disambiguated. One task of formalization simply consists in providing an unambiguous representation of whatever is the subject matter of formalization (language, facts of physics, biology, chemistry,...).

1.3.1. Lexical Ambiguity

Synonymy: Two expressions/lexemes are synonymous if they mean the same.

Homophony: Two expressions/lexemes are homophonous if they coincidentally share the same phonetic form (i.e. are pronounced in the same way).

(38) a. ⟦meat⟧ = parts of (edible) corpse of animal
b. ⟦meet⟧ = encounter

NOTATIONAL CONVENTION: read '⟦α⟧'as the function returning the meaning of α.

(39) a. hare, hair
b. night, knight

(40) Greek
a. $φίλα_{PF}$ = [fila], ⟦φίλα⟧ = kiss!
b. $φύλλα_{PF}$= [fila], ⟦φύλλα⟧ = leaves (on a tree, e.g.)
c. $φύλα_{PF}$ = [fila], ⟦φύλα⟧ = sexes

POLYSEMY: An expression/lexeme displays polysemy iff its denotation consists of more than one related meaning variants.

(41) a. Sam is healthy (i.e. Sam's body is in good condition)
b. Sam's diet is healthy (i.e. Sam's diet improves the condition of his body)
c. Sam's attitude is healthy (i.e. Sam's way of thinking is approved by speaker)

(42) a. Can I have the bat?
b. Instruction to hand over the baseball bat
c. Request for a specimen of a group of animals

(43) a. Das Schloss ist alt
b. The castle is old
c. The lock is old

(44) [katse]
a. *Greek:* ⟦κάτσε⟧ = Sit down!
b. *German:* ⟦katze⟧ = Cat

○ Potential worry: Is *bat* really ambiguous, or just vague/polysemous? After all, many words refer to a variety of things which do not superficially resemble each other (e.g., elephants are animals, but so are certain microbes and corals).

(45) *Scenario: a world with just one elephant and one microbe*
There are just two animals on this goddamn planet! *True statement*

(46) *Scenario: a world with just one Vampire bat and one baseball bat*
There are just two bats on this goddamn planet! *False statement*

Exercises

○ Are the following examples cases of homophony or polysemy or synonymy (more than one answer may apply)? Why?

(47) a. $[\![\text{έρημος}_{N}]\!]$ = dessert
b. $[\![\text{έρημος}_{A}]\!]$ = deserted

(48) a. windows (in: 'Open windows cause draft.')
b. Windows (in: 'Open Windows, and wait for the system to start up!')

(49) a. for you too
b. 4U2

(50) a. [weits] (in 'Tom Waits', [the singer])
b. [weits] (in 'Tom waits for you')

○ If two homophonous expressions are translated into another language, the translations

(51) a. ☐ never
b. ☐ sometimes
c. ☐ always
d. ☐ in principle can

retain homophony. (More than one answer may apply!)

○ Same question as above for polysemy.

1.3.2. Structural Ambiguity (aka amphiboly)

By convention, there is no ambiguity in formal languages such as algebra. The expression in (52)a can - by convention - only be interpreted as in (52)b, but not as in (52)c.

(52) a. $9 \times 2 + 5 =$
b. $9 \times 2 + 5 = (9 \times 2) + 5 = 23$
c. $9 \times 2 + 5 \neq 9 \times (2 + 5) = 90$

1.3.2.1. Ambiguity in overt syntax

A surface string can be assigned more than one parse or syntactic representation (*amphiboly*):

(53) John saw her duck

a. $[_{IP}$ John $[_{VP}$ saw $[_{NP}$ her $[_{N°}$ duck$]]]]$
b. $[_{IP}$ John $[_{VP}$ saw $[_{VP}$ her $[_{V°}$ duck$]]]]$

(54) Save soap and waste paper

a. Save $[_{NP}$ soap$]$ and $[_{NP}$ waste paper$]$
b. $[_{IP}$ Save soap$]$ and $[_{IP}$ waste paper$]$

(55) **AMBIGUITY**$_{\mathrm{DEF}}$

A string α is *ambiguous* if and only if there is a situation in which α is simultaneously evaluated as true and as false.

Exercise

Describe the ambiguity of (56) and (57) by using criterion (55). More precisely, for each example, provide two different situations each of which makes the sentence true in one reading only. Draw a trees for each interpretation.

(56) She's the mother of an infant daughter who works twelve hours a day.

(57) The chicken are ready to eat.

Excursus: Garden path Sentences

So-called *garden path* sentences elicit especially clear evidence for the existence of syntactic ambiguity. In garden path constructions, two factors conspire to yield the effect of anomality: First, only parts of the sentence support two distinct parses. Second, when processing the string, there is a preference to assign the clause the structure which in the end turns out to be unavailable. Thus, the listener/reader has to backtrack, resulting in additional processing load:

(58) Because she jogs a mile doesn't seem difficult.

(59) The boy got fat melted.

(60) While Bill hunted the deer ran into the woods.

(61) The horse raced past the barn fell.

(62) The old man the boat.

1.3.2.2. Ambiguity at LF

In sentence (63), the subject and the object position are occupied by two QUANTIFIER PHRASES (‘QPs’; *some critic* and *every movie*). QPs are formed by combining what semanticists call a common noun (CN), a category with the (syntactic) label ‘NP’, with a quantifier such as *some, no, every, most, few, less than 6,....* (63) possesses now two logically independent readings, which correspond to the two different orders in which the quantifiers *some critic* and *every movie* can be construed, resulting in a so-called **QUANTIFIER SCOPE AMBIGUITY**:

(63) Some critic has seen every movie

a. $\exists x[\text{critic}(x) \wedge \forall y[\text{movie}(y) \rightarrow \text{see}(y)(x)]]$
"There is an x such that x is a critic x and for every y, such that y is a movie, x saw y"

b. $\forall y[\text{movie}(x) \rightarrow \exists x[\text{critic}(x) \wedge \text{see}(y)(x)]]$
"For every y such that y is a movie, there is an x such that x is a critic and x saw y"

Applying the ambiguity test in (55) reveals that there are in fact two different readings associated with a single surface representation.

(64) Scenario which satisfies (63) in reading (63)b, but not in interpretation (63)a:

MOVIE	WAS_SEEN_BY	CRITIC
Vertigo	⇨	Siskel, Ebert
South Park	⇨	Siskel
Kill Bill	⇨	Ebert

Crucially, the inverted scope reading (63)b does not entail the surface scope (63)a, it is logically independent. Thus, both interpretations are called for the representation of sentence (63).

(65) *Question*: Is it possible to construe a scenario in which (63) comes out as true in reading (63)a only?

Examples

Some further examples for ambiguity (at LF and/or in syntax):

(66) a. Which problem did everybody solve?
b. She doesn't eat cheese because it smells.
c. John is looking for a house.
d. They required them to learn only Japanese.
e. All is not right with the American way (Howard Lasnik)
f. What I want to see is that this remains a country where someone can always get rich (George W. Bush)

(67) Wir schwören nicht die Wahrheit zu sagen

a. We do not swear to say the truth ➜ It is possible that we are telling lies
b. We swear not to say the truth ➜ We *are* telling lies

Exercises

The example below are ambiguous. Determine whether the surface interpretation entails the inverted reading, or vice versa, or none entails the other, by completing the relational diagrams.

Instruction for diagrams: For any two nodes a and b, connect a and b with an arrow pointing to b if the first node is related to the second one by the intended relation R, i.e. if $\langle a,b \rangle \in R$.

(68) Not every boy saw a movie

a. Boys Movies
● ○
● ○
● ○

b. Boys Movies
● ○
● ○
● ○

(69) Nobody liked every movie

a. Boys Movies
● ○
● ○
● ○

b. Boys Movies
● ○
● ○
● ○

(70) The boys didn't see a movie

a. Boys Movies
● ○
● ○
● ○

b. Boys Movies
● ○
● ○
● ○

(71) Exactly one boy didn't go to the movies

a. Boys Movies
● ○
● ○
● ○

b. Boys Movies
● ○
● ○
● ○

(72) Exactly half the boys saw some movie (Bonus)

NEXT: Question I: What is the definition of 'meaning'?
➜ introducing the concept of *logical form*, illustrated by inference schemes

Question II: How are natural language expressions assigned meanings?
How do speakers assign meanings to linguistic utterances?

Question III: How exactly do the processes of assigning meaning in natural language differ from the ones developed for formal languages (such as propositional logic or predicate calculus)?

1.4. Logical Form and Inferences

(☞ Objective III of NLS)

Native speakers have intuitions about the correctness of certain inferences. For instance, the following two types of inferences below are held to be correct:

		Name	*Logical Form*
(73)	If John is sick, he can't go to school John is sick ∴ John can't go to school	(*Modus ponens*)	$A \rightarrow B$ A $\therefore B$
(74)	If Sam is hungry, he eats Sam does not eat ∴ Sam is not hungry	(*Modus tollens*)	$A \rightarrow B$ $\neg B$ $\therefore \neg A$

Knowing only the truth or falsity of any given premises or conclusions does not enable one to determine the validity of an inference. In order to understand the validity of an argument, it is also necessary to grasp the *logical form* of the statements involved and their relation to each other. The logical form in this philosophical sense[2] can be thought of as the meaning skeleton of a sentence which is the result of (i) substituting all contentful items for variable symbols that take their place (A and B in the examples above) and (ii) correctly interpreting the logical relations that hold between these symbols in terms of so called *logical constants* ($\rightarrow$, $\vee$,$\wedge$, $\neg$, among others). Take the inference in (73). *John is sick* is a sentence containing a number of contentful items (at least *John* and *sick).* A radical way to get rid of this lexical information is to substitute the whole sentence by a variable over sentences, say A. Similarly, *He can't go to school* can be symbolized as B. The *if* part is traditionally (but incorrectly - see below) rendered by *material implication* (see GAMUT, p.33). Thus, the NL expression in the first line of the left-hand side of (73) can be translated in the *symbolic representation* '$A \rightarrow B$', which will for the moment be taken to be its logical form.

This procedure of formalizing NL into a *symbolic logic* makes it possible to reveal the constant parts of meaning of expressions, i.e. those hidden meaning properties of sentences which do not change even if one exchanges each single lexical item. *Modus ponens* (that's the name traditionally assigned to the scheme in (73)) is for instance *valid* (= correct) for any particular choice of true premises for the value of the variables A and B.

(75)	If the sun shines, Mary wins a car. The sun shines ∴ Mary wins a car	$A \rightarrow B$ A $\therefore B$

In order to have a way to signify whether a statement is true or false, we will furthermore use the symbols 'T' for true and 'F' for 'false, and 'T(A)' and 'F(A)' for 'A is true' and 'A is false',

[2]Philosophical use of the term *logical form* differs substantially from the way is employed in current linguistics. In philosophy, it denotes a disambiguated representation (usually formalized in terms of some type of predicate logic). In linguistics, the term logical form (or LF; first suggested in Chomsky 1976 and May 1977) is used to refer to a syntactic level of representation which connects core syntax to the semantic component (following Chomsky 1976 and May 1977; see http://kleene.ss.uci.edu/~rmay/LogicalForm.html for some details).

respectively. On this view, the connective *if* or *if.... then* is treated as material implication, defined by the truth-table[3] below:

(76) *material implication*

	p	q	p → q
1.	T	T	T
2.	T	F	F
3.	F	T	T
4.	F	F	T

Note that according to the truth table (76), the whole conditional is true whenever the antecedent is false (see rows 3 and 4; this situation is traditionally referred to as *ex falsum quodlibet*/'[infer] from the false whatever you like'). For some cases, this interpretation of NL conditionals in terms of material implication seems to work. Assume that in a given situation, there is a person referred to as *Mary*, Mary is 1,65m tall, and the sun shines in this situation. Then, (77) is a true inference according to the truth-table. Assume, alternatively, another situation, which minimally differs from the first one in that it rains. Again, the conditional can be uttered felicitously (= successfully, correctly) to describe this situation, as shown by (78). Note that the fact that Mary is 1,65 in this situation does NOT logically follow from the falseness of 'the sun shines'. A conditional which is evaluated as true may also be made up of a false antecedent AND a false consequence. It is for this reason, the last line of (78) is not prefixed by the entailment character (∴) which is reserved for logical inferences only.

		Logical Form	*Evaluation*
(77)	If the sun shines, Mary is 1,65m tall	A → B	T(A→B)
	The sun shines	A	T(A)
	∴ Mary is 1,65m tall	∴ B	T(A)
(78)	If the sun shines, Mary is 1,65m tall	A → B	T(A→B)
	The sun doesn't shine	¬A	F(A)
	Mary is 1,65m tall	∴ B	T(A)

Moreover, in the case at hand, this independence of *ex falsum quodlibet* conditionals on the truth of the consequence seems intuitively acceptable, because the weather does not have an influence on Mary's height. Other cases pose problems, though, which will be briefly taken up below.

1.4.1. Classic Syllogistic Logic

The first systematic study of inferences and the logical form of sentences is due to Aristotle (Αναλύτικα πρώτερα - *Prior Analytics,* part of the *Organon*; 4th c. BC), who considered a designated group of inferences, the *syllogisms*:

(79)	Every duck is a bird	(*Barbara*)	∀x[Ax → Bx]	A
	Every bird is an animal		∀x[Bx → Cx]	A
	∴ Every duck is an animal		∀x[Ax → Cx]	A

[3] Truth-tables were first used by Wittgenstein, in his *Tractatus logico-philosophicus* in 1922.

(80) No bird is a reptile (*Celarent*) $\forall x[Ax \rightarrow \neg Bx]$ E
All sea gulls are birds $\forall x[Cx \rightarrow Ax]$ A
$\therefore$No sea gull is a reptile $\forall x[Cx \rightarrow \neg Bx]$ E
(follows from: All sea gulls are not a reptile)

Syllogisms are valid arguments with two *premises* and one *conclusion*. In addition, the three statements contain only three *terms*[4] (in the example above roughly DPs), each of which appears twice. Consider e.g. (79). Here, the three terms are *every duck, a bird* and *an animal,* each of which occurs two times (although sometime in a slightly different guise: cf. *a bird* vs. *every bird*; see below). The argument in such syllogisms is valid by virtue of the fact that it would not be possible to assert the premises and to deny the conclusion without contradicting oneself.

● Examples of an invalid syllogism (the second involves what's called *Goldbach's Conjecture* as the second premis):

(81) Every duck is a bird $\forall x[Ax \rightarrow Bx]$
(At least) one bird is called Donald $\exists x[Ax \wedge Bx]$
Every duck is called Donald

(82) At least one number is an even number greater than two $\exists x[Ax \wedge Bx]$
Every even number greater than two is the sum of two prime numbers $\forall x[Bx \rightarrow Cx]$
$\therefore$Every number is the sum of two prime numbers $\forall x[Ax \rightarrow Cx]$

The conclusion in (82) is invalidated by the existence of numbers such as 3, which cannot be decomposed into the sum of two primes.

1.4.2. The structure of syllogisms

More precisely, in syllogistic, the subject and the predicate of the conclusion each occur in one of the premises, together with a third term (the *middle*) that is found in both premises but not in the conclusion. Take e.g. the conclusion of (79), which consists of a subject term (lets call it α) and the predicate term (which will be referred to as β):

(83) Conclusion: $[_\alpha$ Every duck] is $[_\beta$ an animal]

In this particular case, the α constituent shows up in the first premise (see (84)), while the β-portion of the sentence functions as the predicate of the second premise ((85)). Since every premise (by definition) contains exactly two positions for a term, there are two remaining slots, which are filled by the middle term γ:

(84) 1st premise: $[_\alpha$ Every duck] is $[_\gamma$ a bird]
(85) 2nd premise:$[_\gamma$ Every bird] is $[_\beta$ an animal]

A syllogism argues that because α and γ are related in certain ways to β (the middle) in the premises, they are related in a certain way to one another in the conclusion.

---

[4]*Term* is the Latin version of Greek όρος, meaning 'limit'. Aristotle used the terminus for the subject and the predicate of a syllogism. Today, *term* usually refers to the arguments of a predicate.

1.4.3. Combinatorics I: Figures

Recall that there are all together four positions in the two premises, and there are two terms from the conclusion (α and β) which have to be distributed into these slots. In principle, this generates four different combinatorial options, traditionally referred to as *figures*.

(86)	*1st figure*	*2nd figure*	*3rd figure*	*4th figure*
	α γ	β γ		
	γ β	α γ		
	α β	α β		

That is, there are in principle four ways to relate the middle term to α and γ.

Exercise

Fill in the remaining two figures in (86).

1.4.4. Combinatorics II: Mode

Next, let us turn to the specific shape of the three terms. On the assumption, which is crucial for present purposes, that α, β and γ always stand for exactly the same string in a given syllogism, it must be concluded that they do not represent full DPs, but smaller constituents, which are proper subparts of DPs. In the example (79), e.g., it is only the *bird* part which is kept constant, not the whole DP *a bird* or every bird. Roughly, these portions seem to correspond to NPs. Thus, the remaining parts must be specified: the determiners of the NPs. Traditional (so-called categorical) syllogistic, the study of these structures, allows for the following four options, thereby reducing each premise and each conclusion to one of four basic forms (A and B are here variables over - possibly identical - NPs):

(87)				
	a.	Every A is B	A	(***A**ffirmo*)
	b.	No A is a B	E	(*n**E**go*)
	c.	Some A is a B	I	(*aff**I**rmo)*
	d.	Some A is not B	O	(*neg**O**)*

Respectively, these forms are known as A, E, I, and O propositions, after the vowels in the Latin terms *affirmo* and *nego*.

How many possible syllogisms are there now? Since syllogisms contain three clauses, and each clause can take one of four different forms, there are 64 (= 4^3) different types of syllogisms. Moreover, given that there are four different figures, the overall number of possible syllogism increases to 264. Out of these, only 19 are usually considered to be valid (the exact number depends on the number of syllogisms which are hypothesized to be reducible to more basic syllogisms).

○ For interesting interactive exercises and more info on syllogism have a look at
http://www.phil.gu.se/johan/ollb/Syllog.machine.html

Question: Do the 264 syllogisms, which may vary according to figure and mode, suffice to account for all possible inferences in NL? If not, what else is required?

1.5. THEORIES OF MEANING

Different disciplines concentrate on different aspects of meaning, among them ethnology, sociolinguistics, lexicography, rhetoric, semiotics, pragmatics, neuropsychology (Event Related Potentials), formal semantics. As for linguistics, studies of meaning have followed one (or more) of three major traditions, which focus on different aspects of linguistic meaning and which can be roughly characterized as below:

- *Non-psychologistic/Referential tradition*

  Expressions are symbols referring to (possibly abstract) objects (Frege 1892; Tarski 1935; Montague 1973). The study of meaning is the study of relations between objects and symbols. Semantics is part of mathematics.

  ⇨ The paradigm to be adopted here!

- *Psychologistic/Mentalistic theories*

  Expressions refer to concepts (Fodor 1975; Jackendoff 1983). The study of meaning is the study of how meanings are mentally represented and how they can be manipulated. Semantics is part of cognitive psychology.

- *Pragmatic/Social theories*

  The meaning of expressions is the way in which they are used. (*Ordinary language school*, including Austin, Searle and Grice)

1.5.1. Semantic Competence and Knowledge of Meaning

Formal semantics studies the ability to assign meanings to linguistic utterances. According to the non-psychologistic tradition, initiated by Frege, formal semantics is not concerned with psychological reality, and semantics can be distinguished from knowledge of semantics (Lewis 1975). In contrast, the mentalistics, cognitive, or conceptual approaches towards semantics focus on the way in which meanings are represented in the mind (Fillmore, Langacker, Talmy, Wierzbicka). Drawing the parallel to syntax, Jackendoff (1996) remarks that on the former view, one would be left with the odd result that formal semantics studies E(xternalized)-semantics, while formal syntax studies syntactic competence or I(nternalized)-language (Chomsky 1986):

(88) a. *Chomsky (1986)*

E-language		
I-language	⇨	Object of study of formal syntax

b. *Jackendoff (1996)*

E-semantics	⇨	Object of study of formal semantics
I-semantics	⇨	Object of study of Conceptual Semantics

However, it is also possible to combine the non-psychologistic approach with the psychologically oriented perspective, resulting in a definition of semantics as the study of “mind-internal intuitions of mind-external relations” (Partee 1998).

1.5.2. Theories of Meaning I: Ideational Theories

On one of two competing views about the meaning of NL expressions, words denote ideas (or concepts or thoughts or mental images). For John Locke (1632-1704), one of the main proponents, ideas are the representations of facts and words are the signs (the names) of ideas. Since ideas are mind-internal, they cannot be communicated, requiring words to make them public.

FOUR PROBLEMS:

- What are the exact semantic values? Mental images are too detailed for the meaning, concepts are notoriously ill-defined.
- How does e.g. the idea/concept of 'bird' look like? There is too much diversity in order to define a single picture or concept that fits them all (parrot, flamingo, kiwi, ostrich,...).
- Problem of intersubjectivity: How can a mind-internal idea be made public?
- What do NPs such as *unicorn, Pegasus,* or *Sherlock Holmes*, which lack a referent, denote? Are there ideas for non-existing objects? If yes, there must be an infinite number of such ideas.

Gottfried Wilhelm Leibniz (1646-1716) held a more moderate version of an ideational theory. Ideas and language are related, but not in a one-to-one fashion. Words can be used without knowing their content, i.e. the idea they stand for. This cannot be explained if ideas and words are strictly associate with each other.

1.5.3. Theories of Meaning II: Referential Theories

In its purest form, the referential theory states that the meaning of an expressions is the (possibly abstract) object or individual the expression stands for. On this view, the name *London* denotes e.g. the capital of England. That is, content words refer to objects and individuals, or groups or collections thereof. Plato introduced the concept of 'forms' for common nouns such as *dog*, or abstract nouns such as *justice*, or *exploitation*. The pure referential theory runs into a number of problems, though (for an overview see e.g. Lycan 2000; Taylor 1998; Grayling 1997).

Following Peirce (see also Ogden and Richards (1923)), it has become common to employ the semiotic triangle to describe the relation between linguistics signs (e.g. the word "this dog"), their denotation (e.g. the animal called Fred), and its intension (the concept 'dog'). The meaning of an expression is the relations depicted by the diagram.

(89)

Sign
Signé (Saussure 1916)
Zeichen (Frege 1892)

Referent	*Intension* (Carnap 1937)
Signifié	Signifiant
Gegenstand/Bedeutung (Frege 1892)	Begriff/Sinn (Frege 1892)
Extension (Carnap 1937)	Concept

☟ While referential theories of meaning work well for names and certain noun phrases (*this dog*), they do not capture the meaning of most other linguistic categories such as predicates (*boredom, relatedness*), prepositions (*be between x and y*), quantifiers (*the, some, less than two, only*), and connectives (*and, or, but)*. In fact, it is not even clear what sentences should refer to. (Frege postulated "the True" and "the False" as referents for propositions.)

☟ Structuralist subvarieties moreover suffer from the drawback that they merely *describe* the relation between the ingredients; they fail to provide a precise definition of meaning.

Ideational and referential theories contrast in that they offer different solutions to the questions of which objects represent the meaning of words. They do not necessarily differ in the assumptions they make about what meanings should be assigned to the other basic unit, i.e. sentences, though. Even though there is also a variety of approaches towards sentence meanings, these notes focus on the probably most widely accepted theory only.[5]

1.6. Truth Conditional Semantics

Although most people have never heard sentence (90), and although only a few know whether (90) is true (in fact it is; Gingrich still holds the world record in hand shaking), native speakers of English have clear intuitions as to the conditions under which (90) is true:

(90) On August 22 1998, Newt Gingrich shook 3,609 hands

More precisely, speakers know that the statement in (91) holds, irrespective where, when and by whom the sentence under quotes is uttered:

(91) The sentence "On August 22 1998, Newt Gingrich shook 3,609 hands" is true if and only if Newt Gingrich shook 3,609 hands on August 22 1998.

Following Davidson's implementation of an idea of Tarski's, the meaning of a sentence can be identified with its truth conditions, i.e., the necessary and sufficient conditions for its truth:

(92) To know the meaning of a sentence is to know its ***truth conditions.***

(93) "To understand a sentence, to know what is asserted by it, is the same as to know under what conditions it would be true. . . . To know the truth condition of a sentence is (in most cases) much less than to know its truth-value, but it is the necessary starting point for finding out its truth-value." (Rudolf Carnap, excerpted from ©Britannica 2002 article on metalogic)

(94) Einen Satz verstehen, heißt, wissen, was der Fall ist, wenn er wahr ist.
'To understand a proposition means to know what is the case if it is true'
Wittgenstein (1922), *Tractatus Logico-Philosophicus,* Nr 4.024

[5]This simplified picture does not do justice to the multiplicity of theories of meaning, among them the many varieties of deflational, correspondence and coherence theories.

Semantic theories which equate meaning with truth conditions are also called alethic (gr.αλέιθια/truth), and fall in the group of so-called *correspondence* theories of truth. There are various other ways of thinking about truth, though, among them coherence theories, pragmatic theories and deflationary theories. For introduction see e.g. Lycan 2000, Grayling 1997, Taylor 1998 and the matching entries in the online Stanford Encyclopedia of Philosophy).

1.6.1. Object and Meta language

(95) *Observation*: The words 'dinosaurs', 'writers' and 'Sven' is used in two different ways in the examples below:

(96) a. Dinosaurs are extinct, they died out a while ago
b. 'Dinosaurs' has nine letters
c. One should not use the word 'dinosaurs' to refer to elderly artists
d. As soon as they entered the club, they felt like dinosaurs.

(97) 'Writers' rhymes with 'lighers'

(98) a. Sven has a nice name
b. 'Sven' is a nice name

In (96)a and (96)d, the word is used as *object language*, that is it refers to all the species of extinct animals called dinosaurs. In (96)b and (96)c, the occurrence refers to the string of letters d^i^n^o^s^a^u^r^s which constitutes the word. This is the *meta language* use. Tarski observed that it is essential to distinguish two types of use of language:

Object language: The expressions that one theorizes about (henceforth **bold face**).

Meta language: The language used to theorize about the object language.

NOTATIONAL CONVENTION: Special typography (italics, ...) mark meta language.

Generally, theories of meaning relate expressions of object language to an interpretation in a meta language.

● A linguistically interesting class of meta language uses is represented by *metalanguage negation*. (99)a is a consistent statement. If negation were interpreted as the regular object language connective, (99)a could be decomposed into (99)b and (99)b - two sentences that cannot be true at the same time:

(99) a. This joke is not old - it is very old.
b. It is not the case that this joke is old.
c. It is not the case that this joke is very old.

Rather, negation in (99)a contributes to the sentence meaning something like the claim that the choice of the AP *old* - a meta language use of the expression *old* - is not as appropriate as is the use of the phrase *very old*.

Example

Metalanguage is often used in legal definitions. The following section is from the *Drivers Handbook and Examination Manual for Germany,* issued by the Military Police:

> Traffic laws require users of public roads to conduct themselves so that no person is endangered, injured, impeded, or unreasonably inconvenienced when the inconvenience could have been avoided under the given circumstances.
> a. "Users of public roads" refers to drivers of motor vehicles, bicyclists, pedestrians, horseback riders, drivers of animal carts, and owners of domestic animals who allow their animals to stray onto public roads.
> b. "Unavoidable circumstances" are those over which drivers have no immediate control (for example, unavoidable noise or exhaust fumes produced by heavy traffic).
> c. "Avoidable circumstances" are those over which the driver has immediate control (for example, squealing tires in residential areas, racing the motor, honking the horn, playing loud music).
> d. "Endangering" means to place other users of the road in danger by
> (1) Failing to obey traffic signs.
> (2) Failing to yield the right-of-way.
> (3) Failing to warn approaching traffic that a vehicle is disabled or parked on the highway...........
> (8) Driving when fatigued, ill, or under the influence of alcohol or drugs.
> e. "Injure" is to cause physical harm to other users of the road or damage property. Splashing mud or water on pedestrians is in this category.
> f. "Impede" means to fail to adjust driving speed to the flow of traffic or blocking other traffic when parked.
> g. "Inconvenience" is to make more noise than necessary (for example, honking the horn, playing loud music, driving with a faulty exhaust system, racing the engine, letting the engine idle for more than 30 seconds

Exercise

Which of the sentences are false? How can they be turned into true statements? Assume that meta language expressions are marked graphically, as stated in (98)b (from Heim & Kratzer 1998):

(100) a. Boston is a big city
b. Boston has six letters
c. Boston is the name of a city
d. Boston is the name of Boston
e. Boston might have been called New York

1.6.2. The Vehicles of Meaning

(101) *Hypothesis A:* Meanings correspond to physical objects which are directly related to sentences. Meanings are e.g. reflexes of sound waves or mental states.

(102) *Problem I:* Meaning of one and the same sentence may differ, depending on the situation and the people who use it.

(103) a. I am cold and freezing
b. They are fantastic
c. Stop!

(104) *Problem II:* Sentences in different languages may express the same content.

(105) a. I do not want to read it
b. Δεν θέλω να το διαβάσω
c. Je ne veux pas le lire

(106) *Hypothesis B:* The meaning of a sentence is to be located in the *idea* it is associated with.

(107) *Problem III:* Ideas are individually internalized constructs, just like pain, hunger or love. How can they be shared by the speakers of the language community? How is intersubjectivity ensured?

(108) *Hypothesis C:* The meaning of a sentence is encoded in an abstract object, the *proposition.*

(109) *Solutions to problems I-III*

I. One and the same sentence may be associated with different propositions, depending on the situation in which the sentence is uttered (for formalization see 1.6.3. below).

II. Different sentences may express the same proposition (in one and the same language or in different languages).

III. Propositions have ontologically autonomous status, they are abstract objects. Hence, they can be internalized by different individuals.

(110) *Some Consequences*

- Propositions, and not sentences, are true or false.
- Not sentences, but propositions enter into logical relation such as inferences (i.e. propositions entail or contradict one another).
- Propositions are abstract (Platonic) objects, which are intersubjectively verifiable.
- On the view propagated above, the universe is inflated in that it is full of invisible, non-detectable platonic creatures (this view leads to *Metaphysical Realism*).

1.6.3. Davidsonian Semantics

The next goal consists in spelling out Davidson's program of a Tarski semantics for natural language (which also follows very closely Frege's ideas, even though Davidson seem not to have been aware of Frege's work at the time he developed his theory). As the discussion proceeds, some new concepts will be introduced, most of which aid in rendering the intuitions behind the Tarski-Davidsonian semantics more clearly. To begin with, it is useful to define a number of conventions.

NOTATIONAL CONVENTION: For any α, ⟦α⟧ is the *denotation* or *semantic value* of α.

The interpretation function ⟦⟧ maps expressions of the object language (in this case English) to their meanings, it assigns these expression a semantic value.

NOTATIONAL CONVENTION: Read ‘iff’ as ‘if and only if’

(111) *Assumption:* For any sentence φ, the truth conditions for φ are either 1 (T) or 0 (F).

○ The core of Tarski’s account is contained in his Convention (T):

(112) *Tarski’s Convention (T)*
For any sentence φ, φ is true in a language L if and only if p

The t-conditions for a classic example accordingly are as follows:

(113) ‘Snow is white’ is true iff snow is white

Tarski’s actual goal was to give a formally correct definition of what it means to be ‘true’, i.e. the truth predicate. So he was not primarily concerned with the question of what sentences (or propositions or formulas) mean. Moreover, he explicitly renounced the possibility of applying his method, which was designed for formal languages, to NL, because he thought that NL were to irregular and inconsistent. Davidson’s and other’s work in the 1960ies showed otherwise.

On Davidson’s re-interpretation of Condition (T), the t-conditions of NL-sentences can be collected by a simple procedure: Take the object language sentence, and specify that it is true, relative to a situation, just in case the meta language version of the sentence (which one uses to describe the world) is true in that situation. Reference to situations is motivated by the need to

(114)

***Truth Conditions for Sentences** (Davidson)*

For any sentence φ and situation s: $[\![\boldsymbol{\varphi}]\!]$ = 1 in s iff φ is true in s.
(left-hand side φ is object language, the rest is meta language)

According to Convention (T), every sentence has truth conditions which define the requirements that a sentence must satisfy for being evaluated as true. That is, a theory of meaning is concerned with the meaning of the *object* language expressions in a given language, and uses *meta* language to describe these meanings.

Example

Applying the general format of Davidson’s version of Convention T leads to biconditionals (‘iff’, symbol ‘↔’) in which the left-hand side of the biconditional is occupied by a object language formula, while the right hand side spells out the t-conditions in meta language. In many cases, object and meta language coincide, as e.g. whenever people discuss properties of English by using English. The contrast between object and metalanguage becomes more obvious once the language under study is not English, but another language such as French:

(115) For any s, $\underbrace{[\![\textbf{La neige est blanche}]\!]}_{\text{object language version of } \varphi \text{ (not '}[\![\text{' and '}]\!]\text{'!)}}$ = 1 in s iff $\underbrace{\text{snow is white}}_{\text{metalanguage description of } \varphi}$ in s

NB: Everything except for object language is metalanguage, which includes *(quasi) formal language* that is used to describe the relation between the two occurrences of φ

● Note that we e.g. know now the *conditions* under which (90) - repeated below - is true, but nothing has been said yet about the actual factual content of the statement. Is (90) true in the actual world, or not? That is, a method for computing the actual truth value of sentence is still missing.

(90) For any s, ⟦**On August 22 1998, Newt Gingrich shook 3,609 hands**⟧ = 1 in s iff
On August 22 1998, Newt Gingrich shook 3,609 hands in s

NB: The situation s is here already specified as far as the time parameter goes. This indicates that s can in some cases also be extracted from the object language content of the sentence.

The solution is rather simple, though (it is an instance of what will be called *Function Application* below):

(116) *Computing Truth Values:* Truth values are the result of applying truth-conditions to situations/worlds.

<u>Example:</u>

(117) 1. ⟦**On August 22 1998, Newt Gingrich shook 3,609 hands**⟧ = 1 iff
On August 22 1998, Newt Gingrich shook 3,609 hands

2. In situation s_o (the actual world we live in), it is a fact that on August 22 1998, Newt Gingrich shook 3,609 hands.
In situation s_1, which corresponds to the fictitious world of the Simpsons, such an event did not happen.

3. Evaluated at s_o, the proposition is therefore true, while it is false at s_1.

How sentences express meanings,
and how the latter relate to the world

Truth-conditions on p determine meaning of φ

φ denotes proposition p — applying t-conditions to situation σ yields truth value

A utters sentence φ — φ relates to σ, and thereby to reality/world

Speaker A (plus some thought/idea/intention)

1.6.4. Convention T and the Liar Paradox

Tarski hypothesized that a language must not contain its own truth predicate; otherwise, anomalies generating paradoxes such as in (118) (*Liar Paradox*, discussed by Eubulides of Miletus, 4th c. BC, member of the Megarian school) cannot be avoided:

(118) a. This sentence I am uttering right now is not true.
b. Does a man who says that he is now lying, speak truly?

For Tarski, the Liar Paradox emerged because the predicate 'is false' (or, equivalently, 'is not true') is treated as being part of the object language. But the truth predicate can only be used as a metalanguage, in order to express that something in the object language is true. On this view, the paradoxical sentence *This sentence is false* is no more a sentence of object language, than *Snow is blanche* is a sentence of English. In both cases, the predicate is not part of the object language, and the resulting sentences are therefore ill-formed:[6]

(119) *This sentence is false
(120) $\underbrace{\text{*Snow is}}_{\text{Object language}}\quad\underbrace{\text{blanche}}_{\text{Not part of object language}}$

● Does Tarski's restriction that no language contains its own truth predicate also extend to the following, earlier version of the Liar Paradox? If yes, show why:

(121) All Cretans are liars (Epimenides of Crete, 7th c. BC; Ψευδόμενος)

● What about the following example. Can it be captured by the method described by Tarski?

(122) The next sentence is false. The last sentence is true.

1.7. COMPOSITIONALITY/FREGE'S PRINCIPLE

What makes the procedure of assigning meta language meanings to object language expressions in (114) - repeated as (123) - a non-trivial statement? (123) raises at least three puzzles:

(123) For any situation s and any sentence φ:
$[\![\boldsymbol{\varphi}]\!] = 1$ in s iff φ in s

(124) *Puzzle I:* ● In general: Why are the t-conditions of a sentence not substitutable *salva veritate* (i.e. if they preserve truth)?
○ Illustration: Why does *Grass is green* not mean *Snow is white,* e.g?

[6]Tarski's solution represents only one of many to the Liar Paradox; see e.g. John Barwise & John Etchemendy. 1992. *The Liar*.

(125) a. *Grass is green* is true in s if and only if grass is green in s
b. Grass is green in the same situations in which snow is white
c. ∴ *Grass is green* is true in s if and only if snow is white in s

Analysis: ● In general: The t-conditions have to define the *intension*, not just an extension.
○ Illstration: It does not hold for ***any*** situation, that *Grass is green* means the same as *Snow is white*. In worlds without chlorophyll, e.g., the two sets of situations are not identical.

(126) a. For ***any*** situation s and any **φ**, ⟦**φ**⟧ = 1 in s iff φ in s
b. ⟦**φ**⟧ = {s| φ in s}

{s| ....s....} reads as: 'set of ***all situations,*** such that φ holds in s'

NB: The issue of intensionality will be ignored in what follows.

(127) *Puzzle II:* ● In general: Why can the t-conditions of a sentence not be expanded to include tautologies which are unrelated to the sentence?
○ Illustration: Why does *Grass is green* e.g. not mean grass is green and 12/7 = 1.7143? Then, from (129)a one should be able to infer (129)b. More generally, (129)a should entail that John knows all true statements of mathematics and logic - not a very plausible result.

(128) a. *Grass is green* is true is s if and only if grass is green in s
b. Grass is green in the same situations in which grass is green and 12/7 = 1.7143
c. ∴ *Grass is green* is true in s if and only if grass is green in s and 12/7 = 1.7143 in s

(129) a. John knows/believes that grass is green
b. ¬∴ John knows/believes that grass is green and that 12/7 = 1.7143

Analysis: The content of the t-conditions must be systematically connected to the content of the object language sentence. This can be achieved by translating sentences in a step-by-step fashion, starting with the meaning of the words, and proceeding to the meaning of more complex parts (for details see below). Spurious (= superfluous) conditions in the t-conditions can then be excluded, because they do not correspond to linguistic material in the object language.

➜ NB: This solution connects Puzzle II to the group of puzzles below, which are more linguistically - as opposed to philosophically - motivated.

(130) *Puzzle III:* Any language contains a potentially infinite number of sentences, and any competent speaker can interpret these sentences according to (114) (*salve* restrictions on memory and computational complexity). Thus, speakers would have to acquire a potentially infinite number of correspondence rules between sentences and their truth conditions (which can be thought of as the situations in which the sentence is true).

(131) *Puzzle IV:* ● In general: The interpretation procedure (114) does not assign meanings to expressions that do not denote truth-values.

○ Illustration: Names ('Mr. President!') or predicates ('She knows the difference between *multifarious* and *manifold*') have meaning also in isolation.

(132) *Puzzle V:* ● In general: Many linguistic phenomena indicate that semantics has to be able to 'look into' parts of the clause.

○ Illustration: NPI-licencing and the contrast between (20)c and (21)c, repeated below. Both clauses denote propositions. The contrast cannot be due to contrast in force (universal vs. existential), otherwise, (21)b should be well-formed.

(20)c Everybody who has ever heard about it likes it.
(21)c *Somebody who has ever heard about it likes it.

(21)b *Everybody has ever heard about it.

Thus, natural language semantics needs to provide an account of how the meaning of sentences is derived from the meaning of their parts and which semantic contribution the parts of a sentence make to the denotation of the whole clause. The requirements for a theory of meaning in natural language are much the same as the ones which hold for algebra, which offers algorithms for the computation of operations on numbers. To know an algebra means to know how a finite set of operations over an alphabet (numbers) is interpreted. This includes e.g. the knowledge that the string *2 x (3+9)* is interpreted as 'add 3 to 9 and multiply the result by 2':

(133) *Syntax* *Semantics*

```
        x                 24
      /   \             /    \
     2     +           2      12
          / \                /  \
         3   9              3    9
```

Similarly, to know semantics means to know the operations which allow to compute the meaning of sentences from their recursive part-whole structure provided by syntax:

(134) *Syntax* *Semantics*

```
        IP                   ?
      /    \               /   \
   John     VP          John    ?
           /  \                / \
         met   Mary           ?   Mary
```

> ***Principle of Compositionality (Frege)***
> The meaning of an expression is a function of the meanings of the parts of the expression and of their mode of composition.

Thus, the (modified) Convention T in (114) needs to be supplied by recursive principles which derive the meaning of the sentence from the meaning of its parts.

It is important to observe that compositionality in natural language semantics is not considered as an axiom, or an *a priory* truth, but that the principle serves as a heuristics, i.e. a guideline in the search for empirical insights and generalizations. Compositionality in this sense can for instance be used as a metric for evaluating different theories in that a theory which observes compositionality is preferred over a theory with equal expressive which does not do so.

<u>Exercise</u>

Which type of evidence would invalidate the assumption that meaning is computed compositionally?

The comparison with algebra may also serve to highlight another aspect of formal languages: Consider a natural language expression and its formalization:

(135)	a.	Natural language:	three and nine
	b.	Formal language:	3 + 9
	c.	Interpretation:	12

(135)a is the output of 'syntax' in the linguistic sense, while (135)b can be thought of as its formalization which represents the output of semantics. But the formalized language is subject to syntactic restrictions of its own. For instance, in standard algebras, *3+9* is a well-formed expression, while *39+* is not. Thus, the + operator has a syntax as well as an interpretation (semantics).

In sum, the translation procedure looks as follows: in course of the semantic computation, natural language strings are translated into a formal language. This formal language has - just as natural language - a syntax and a semantics, which serves as the basis for interpretation of the string by semantic rules.

<u>How to choose principles of composition?</u>

Question: What form should the principles of composition take? One per construction?

We know that there is no situation whatsoever which renders the clauses in (136) true, they are contradictions:

(136) a. The black dog slept and the black dog didn't sleep
b. The black dog slept and the dog which is black didn't sleep

(136)a demonstrates that speakers are capable of assigning meanings to strings in a systematic way: adding *didn't* in the right place results in a negative statement. (136)b reveals another curious property of language: in fixing the meaning of the subject, (136)a and (136)b employ two different strategies (attributive AP vs. relative clause) which nonetheless yield identical meanings. That is, the same thought can be expressed in more than one way (reverse of LF-ambiguity, where one string is assigned two readings). It seems plausible that in (136)b, attributive adjectives (*black* in *black dog*) can be assigned the same meaning as relative clauses (*which is black*). Then, both constructions can be analyzed by a single composition rule (♦), instead of having to employ two different principles. Thus, the theory schematized in (137) is to be preferred over the one in (138).

(137) a.

Syntax				*Semantics*	
A	♦	B		⇨	S
		B ♦	C	⇨	S

b. Meaning of A equals meaning of C
A combines with B in the same way as B combines with C

(138) a.

Syntax				*Semantics*	
A	♦	B		⇨	S
		B ■	C	⇨	S

b. A and C differ in meaning
A combines with B by ♦ and B combines with C by ■ (and ♦ ≠■)

Generally, a theory should be construed with the smallest number of ingredients possible.

2. MATH & LOGIC BACKGROUND

2.1. SETS AND RELATIONS

(1) **Set**: An unordered collection of elements (Cantor, Bolzano)

"Unter einer Menge verstehen wir jede Zusammenfassung M von bestimmten, wohlunterschiedenen Objekten M unserer Anschauung oder unseres Denkens (welche die Elemente von M genannt werden) zu einem Ganzen"

a. (Extensional) definition by listing

Example: A = {1,2,3}

b. (Intensional) definition by (Set) **abstraction**

Example: B = {n∈N°| 6>n>0})

(2) **Relations between sets**

a. **Subset**

X ⊆Y if all members of X are members of Y

Example: A ⊆ B

b. **Intersection**

X ∩ Y= {x|x∈X and x∈Y}

Example: A ∩ B = {1,2,3}

c. **Set Union**

X ∪ Y= {x|x∈X or x∈Y}

Example: A ∪ B = {1,2,3,4,5}

(3) **Ordered pair:** Ordered collection of two elements

Example: <1,2>

(4) A **two-place relation** is a set of ordered pairs

Example: {<1,2>,<2,3>,<1,3>})

(5) A relation f is a **function** iff for any x,
if f(x) = y and f(x) = z then y = z

(alternatively: iff for any x, y, z, if <x,y> ∈f and <x,z>∈f, then y = z)

(6) A function f maps members of the set of the domain D to members of the range R ("f is from D to R")

a. **Domain** ('Definitionsbereich')

For any function f, the domain of f is {x|there is some y such that f(x) = y}

(alternatively: {x| there is some y such that <x,y> ∈f})

b. **Range/codomain** ('Wertebereich')

For any function f, the range/codomain of f is {x|there is some y such that f(y) = x}

(alternatively: {x| there is some y such that <y,x> ∈f})

Notational convention: The domain and range for which a function is defined is expressed by the following notation: 'f: D → R' (or 'R^D')

(7) **Function application** (application of a function to its argument)

For any function f and any x in the domain of f,

f(x) := the y, such that <x, y> ∈f ("f applies to x", "f maps x to y")

(8) **Conversion**
$a \in \{x \mid P(x)\}$ iff P(a)

(9) **Cartesian product**
DxD is the set of ordered pairs of elements of D

Example: D = {a,b,c}
DxD = {<a,b>,<b,c>,<a,c>,<b,a>,<c,b>,<c,a>}

(10) **Powerset** ('Potenzmenge')
$\wp$(D) is the set of all subsets of D

Example: D = {a,b}
$\wp$(D) = $\{\varnothing, \{a\}, \{b\}, \{a, b\}\}$

Some formal Properties of Relations

(11) a. R is **symmetric** iff for every x and y: if $\langle x,y\rangle \in R$ then $\langle y,x\rangle \in R$
b. R is **asymmetric** iff for every x and y: if $\langle x,y\rangle \in R$ then $\langle y,x\rangle \notin R$
c. R is **antisymmetric** iff for every x and y: if $\langle x,y\rangle \in R$ and $\langle y,x\rangle \in R$ then x=y

Example: $\{\langle 1,1\rangle\}$: antisymmetric, but no asymmetric (since reflexive)

(12) a. R is **reflexive** iff for every x, $\langle x,x\rangle \in R$
b. R is **irreflexive** iff there is no x such $\langle x,x\rangle \in R$

Example: $\{\langle a,a\rangle,\langle a,b\rangle\}$: non-reflexive but not irreflexive

(13) R is **transitive** iff for each x,y and z: if $\langle x,y\rangle \in R$ and $\langle y,z\rangle \in R$ then $\langle x,z\rangle \in R$

Examples

● Relational nouns denote relations:

(14) a. Homer is the neighbor of Flanders.
b. Neighbor-of relation N = {<Homer, Flanders>, <Flanders, Homer>}
N = $\{\langle x,y\rangle \mid$ x is the neighbor of y$\}$

● Symmetry properties of predicates can explain certain inferences. In all situations in which (15)a is true, (15)b is also true and vice versa.

(15) ⟦**Some Swedes are left-handed singers**⟧ =
= ⟦**Some left-handed singers are Swedes**⟧

The same does not hold for the pair under (16): it could very well be that all Swedes sing and are left-handed, while there being French with the same qualities:

(16) ⟦**All Swedes are left-handed singers**⟧ ≠
≠ ⟦**All left-handed singers are Swedes**⟧

Native speaker know that certain inferences are licit or illicit because they know the meaning of determiners such as *some* and *all*, and the way they combine with the rest of the sentence. More generally, they know that all utterances of the form "some φ are ψ" are logically equivalent to "some ψ are φ", because some is a *symmetric* determiner, while utterances of the form "all φ are ψ" are not logically equivalent to "all ψ are φ" (of course, the latter *can* mean the same in certain situations; equivalence of meaning is said to be *contingent* in these cases).

<u>Exercise</u>

(i) What are the set-theoretic properties of the following relations (are they reflexive, transitive, etc....)?
(ii) Which of them are functions?
(iii) Add a relation which is symmetric, reflexive and transitive.

(17) a. mother_of
b. employer_of
c. older_than
d. located_between_Prag_and_Brunn
e. meet

2.2. PROPOSITIONAL CALCULUS

2.2.1. Truth tables

The *propositional calculus* (PC; also called *statement logic*) is a formal language, which can be used for representing certain aspects of natural language (NL), such as inferences which do not involve quantifiers (*every, some,....*). Just like NL, the formal language PC contains operations which make it possible to generate an infinite set of new sentences out of a (possibly) finite set of basic expressions. And just like NL, statement logic achieves this by using recursive rules.
Turning to the details, PC consists of two parts:

(i) a syntax which defines the set of well-formed formulas (or wff's), and
(ii) a semantic component which provides the interpretation of these formulas.

Again, this is the same what we find in the analysis of NL.[7]

The special property of statement logic, which sets it apart from other languages - and therefore also from NL - is that it only contains statements: There are no nouns, predicates, adjectives or any other categories. One usually expresses this by saying that the only syntactic category of statement logic is that of a *formula* - or, to be precise, a *well-formed formula* (wff). Wff's are recursively defined by the following syntactic rules:

[7]In logic, a calculus is a formal system that derives the logical entailments and all wffs of a language by rules that only specify the *syntax* of the expressions, without reference to the meaning, the *semantics*, of the symbols.

(18) *Syntax*: a recursive definition of the set of all well-formed formulas

a. *Basic clause:* every atomic (i.e. non-composite) statement is a well-formed formula

b. *Recursion clause:*

i. If φ is a wff, then so is $\neg\varphi$ (negation, *not* φ)

ii. If φ and ψ are wff's, then so are:

$[\varphi \wedge \psi]$ (conjunction, φ *and* ψ)
$[\varphi \vee \psi]$ (disjunction, φ *or* ψ)
$[\varphi \rightarrow \psi]$ (material implication, also called conditional, *if* φ *then* ψ)
$[\varphi \leftrightarrow \psi]$ (material equivalence, or biconditional, φ *if and only if* ψ)

These rules yield for instance wff's like '$[[\neg p \rightarrow m] \vee \neg q] \rightarrow \neg[\neg k \vee l]$'. In order to understand the meaning of these strings, the syntax has to be supplied with a semantic component. The semantics is standardly provided in terms of truth-tables (t-tables; the one below combines 5 t-tables into one)

(19) *Semantics*:

	p	q	¬p	p ∨ q	p ∧ q	p → q	p ↔ q
	T	T	F	T	T	T	T
☞	T	F		T	F	F	F
	F	T	T	T	F	T	F
	F	F		F	F	T	T

Read as follows: Pick one of the four rows (e.g. the second one ☞). Then, the value for p is as given in the first column (in our case T), and the value for q is as in the second one of the same row (F). Moreover, the value of a complex formula (such as e.g. $p \wedge q$) can be found in the same row, below the formula which contains the desired connective (in the case at hand $\wedge$). (Note on the side that for negation, only two values are relevant, because $\neg$ only combines with a single statement.)

On how to decompose and interpret complex formulas see e.g. GAMUT, p. 35ff

Exercise Inferences

(20) If the subject has not understood the instructions or has not finished reading the sentence, then he has pressed the wrong button or has failed to answer. If he has failed to answer, then the timer hasn't stopped. The subject has pressed the right button, and the timer has stopped. Therefore, the subject understood the instructions.
(from Rajesh Bhatt, University of Texas 2003, handout on inferences)

2.2.2. Differences between Connectives In PC and NL

Boolean and Non-Boolean *'and'*

- Sentential connective:

(21) Sam is driving and he is listening to the radio

- VP-connective:

(22) Sam is driving and listening to the radio

● NP/DP connective (*Non-Boolean and*):

(23) Sam and Mary are driving

(24) Sam is driving a car and a bike

● Non-commutativity: temporal/causal ordering

(25) Sally drank the medicine and she got sick

(26) Sally got sick and she drank the medicine

<u>Inclusive and Exclusive *'or'*</u>

● Inclusive *or* (the one from the t-table):

(27) Sam is driving or he is listening to the radio (or he's doing both).

● Exclusive *or* (like *either-or*):

(28) Sally is in Athens or she is in Megara
(= If Sally is in Athens, she is not in Megara: [Stoic logician Chryssipus])

<u>Exercise</u>
Define the meaning of exclusive *or* by reducing its meaning to that of a combination of the the other connectives.

Excursus: problems for the interpretation of conditionals
In some cases, the *ex falso quodlibet* clause for material implication (i.e. a formula $p \rightarrow q$ is true whenever p is false) yields the correct result for the interpretation of NL conditionals:

(29) *Intuitively, F(p) and T(q) result in true conditional:*
If a number can be divided by 10, it is even

In (29), a false antecedent can intuitively be combined with a true consequent, yielding a conditional which is evaluated as true, because there are numbers which cannot be divided by 10 (e.g. 2), which are still even. Similarly for (30), which does not demand that one actively avoids an "A" (von Fintel 2000) - getting an "A" would also lead to a reward.

(30) *F(p) and T(q) result in true conditional:*
If you get a "B" on your next history test, I will give you $5.

But in other cases, this combination of truth values does not seem to reflect native speaker's intuitions about the meaning of the *if...(then)* construction:

(31) *F(p) and T(q) yield false conditional*:
Every citizen shall be granted the right to vote, if he/she has reached the age of 18

Assuming that (31) describes the law, the conditional is interpreted to be false if the antecedent is false. Otherwise, the law would leave it open whether citizens under the age of 18 have the right to vote or not. The intended interpretation is $(p \rightarrow q) \wedge (\neg p \rightarrow \neg q)$, which is equivalent to $p \leftrightarrow q$. In a similar vein, statements like (32) are judged to be false, even though they should come out as true irrespective of how the world is structured (given that the contradiction 2 + 2 = 5 is always false):

(32) *F(p) and T(q) yield false conditional*:
If 2 + 2 = 5, then 2 + 2 = 4.

Problems such as these (some of which have been known since antiquity) motivated a revision of the idea that the NL conditional is to be translated as material implication (see in particular work by Angelika Kratzer, David Lewis and Kai von Fintel). These revisions also attempt to capture another observation, i.e. the generalization that conditionals seem odd if the antecedent and consequent are not in some way causally connected (Frege also mentions cases like these):

(33) #If the sun rises in the east, Paris is in France

2.2.3. Tautologies, contradictions and contingencies

The three statements below differ in quality:

(34) Sally is sick and Bill is sick

(35) Sally_k is sick or she_k isn't sick

(36) Sally_k is sick and she_k isn't sick

While the truth of (34) depends on the situation and the way the world is at the temporal slice referred to as 'today', native speakers know that (35) cannot but be true, irrespective of the actual facts in a given situation. (34) is called a *contingent statement*, because its truth is dependent (= contingent) upon language external factors. (35) is a *tautology* (representing the so-called *law of the excluded middle,* because it states that every statement is either true, or false, but nothing inbetween). (36) can finally never be understood as a true statement (at least given that one interprets the two occurrences of *sick* to mean exactly the same in both conjuncts), and exemplifies a *contradiction*.

Whether a formula is a contingency, a tautology or a contradiction can be easily determined from looking at the t-table:

(37)

		Contingency	*Tautology*	*Contradiction*
p	q	$p \wedge q$	$p \vee \neg p$	$p \wedge \neg p$
T	T	T **T** T	T **T** F	T **F** F
T	F	T **F** F	T **T** F	T **F** F
F	T	F **F** T	F **T** T	F **F** T
F	F	F **F** F	F **T** T	F **F** T
		(= (34))	(= (35))	(= (36))

A compound statement is contingent if there is at least one T as well as at least one F beneath its connective in every row of its truth table. It is a tautology if there is a T beneath its connective in every row of its truth table, and a contradiction if all rows are filled by F.

(38) Some tautologies:

a. $p \rightarrow p$
b. $(p \wedge q) \rightarrow p$
c. $p \rightarrow (p \rightarrow q)$ (*Modus ponens*)
d. $\neg p \rightarrow (p \rightarrow q)$ (*ex falso quodlibet)*

2.2.4. Logical Equivalence

Moreover, the t-tables also allow one to see which two formulas are equivalent (symbolized by $\Leftrightarrow$). Two statements are logically equivalent if they have the same truth values regardless of the truth values assigned to their atomic components. For instance, it turns out that '$p \rightarrow q$' is equivalent to '$\neg p \vee q$', and that '$\neg\neg p$' is equivalent to 'p':

Note: Logical equivalence is not the same as material equivalence (symbol $\leftrightarrow$) - even though the two concepts are systematically related one to the other. Logical equivalence is a property that two formulas can have, while material equivalence is a connective, which is part of a formula. More on that below (meta vs. object language).

(39)

p	q	p	→	q	¬p	∨	q	p	¬ ¬ p
T	T	T	**T**	T	F	**T**	T	T	**T** F T
T	F	T	**F**	F	F	**F**	F		
F	T	F	**T**	T	T	**T**	T	F	**F** T F
F	F	F	**T**	F	T	**T**	F		

These findings correspond to the intuitive judgements one has about pairs such as follows:

(40) a. If you are sick, you can stay at home
b. You are not sick, or you stay at home

(41) a. She is happy
b. It is not the case that she is not happy

(42) Some equivalences

a. $p \Leftrightarrow p$
b. $p \wedge q \Leftrightarrow q \wedge p$ (*Commutativity*)
c. $\neg(p \vee q) \Leftrightarrow \neg p \wedge \neg q$ (*De Morgan Law*)
d. $\neg(p \wedge q) \Leftrightarrow \neg p \vee \neg q$ (*De Morgan Law*)
e. $p \rightarrow q \Leftrightarrow \neg p \vee q$ (*Conditional Law*)
f. $p \rightarrow q \Leftrightarrow \neg p \rightarrow \neg q$ (*Contraposition*)

Example of a Deductive Proof

Since equivalent statements have the same t-values, they can be substituted for one another. This makes it possible to derive formulas by means of a *deductive proof*. For instance, contraposition (i.e. (42)f) can be deductively proved from commutativity and (42)e. Crucially, in such a proof, each line is logically equivalent with the preceding one:

(43) To be proved: $p \rightarrow q \Leftrightarrow \neg p \rightarrow \neg q$

1. $p \rightarrow q$	
2. $p \rightarrow q \Leftrightarrow \neg p \vee q$	by (42)e
3. $\neg p \vee q \Leftrightarrow q \vee \neg p$	by commutativity
4. $q \vee \neg p \Leftrightarrow \neg q \rightarrow \neg p$	by (42)e
5. $\neg q \rightarrow \neg p \Leftrightarrow p \rightarrow q$	since line 1 is equivalent with line 4

Exercise

○ Determine whether the pairs are logically equivalent or not by using a t-table:

(44) a. $\neg\neg p \vee q$ b. $\neg q \rightarrow p$

(45) a. $q \vee (q \rightarrow p)$ b. $(q \vee q) \rightarrow p$

(46) a. $\neg(k \wedge \neg(l \rightarrow k))$ b. $\neg(\neg l \vee \neg k)$

(47) a. $r \rightarrow \neg(p \leftrightarrow \neg r)$ b. $\neg r \vee p$

2.2.5. Compositionality and NL

Compositionality is one of the fundamental assumptions about the way in which languages - they may be formal or natural - are interpreted. The reason why compositionality is so important is a different one for formal and for natural languages, though. For formal languages, compositionality is a defining property, it falls out from the way in which the logic is set up. In statement logic, e.g. the t-tables are given in such a wa, that the meaning of a complex formula cannot be anything else but a function of the meaning of the parts. Thus, the meaning of the complete formula always depends systematically on the meaning of the parts. In NL, on the other hands, things are quite different, because the study of NL is empirically oriented (the theory must conform with given data) and the question whether NL is compositional or nor therefore constitutes an empirical issue. Following Frege, most researchers assume compositionality as a guiding principle (a *heuristics*), but there are numerous constructions for which it is not clear yet whether they can be given a compositional analysis.

For further discussion of issues of compositionality see e.g. Kai von Fintels course:

http://semantics-online.org/semantics/

2.3. Quantificational Predicate Logic

2.3.1. Universal Statements

● Sentences in (48)a and (48)b are synonymous - they mean the same in the sense that every situation which makes (48)a true also makes (48)b true, and v.v.

(48) a. If something$_2$ is a bird, then $\left\{\begin{array}{c}\text{it}_2\\ \text{this something}\end{array}\right\}$ lays eggs

b. Every birds lays eggs

● Two components are relevant for capturing the meaning of both sentences:

(i) A logical connective

(ii) Variables

○ First, both involve a *conditional* statement, which can be translated in terms of material implication. This is obvious for the conditional, which is translated as in (49).

(49) a. Natural Language: *If* A *then* B

b. Translation into Statement Logic: A → B

But also sentences with a universal quantifier (*all* or *every*), schematized in (50), can be paraphrased by a statement involving material implication:

(50) a. *All* A (are) B

b. $\underbrace{\text{For all things:}}_{\textbf{QUANTIFIER}}\ \underbrace{\text{if this thing is an A, then this thing is also}}_{\text{A Condition within the }\textbf{SCOPE}\text{ of the quanifier}}$ B

c. For all things: A(this thing) → B(this thing)

○ Second, the quantifier introduces objects, and states for how many of these objects the condition inside its scope has to hold. In our case, *every* demands that all objects satisfy the condition. Crucially, this requirement is not assessed for all objects at the same time, but is computed in a step-by-step fashion by checking for each object, whether this object satsifies the condition or not. This procedure can be compared to inspection of eggs for intergrity, where one picks the eggs one by one, determining whether the actual egg one is checking at the very moment is intact or not. Once one egg has been evaluated, one moves on, picking another one, repeating the procedure, and so on. Similarly, the quantifier can be thought of as giving an instruction to pick each object under consideration, evaluating whether this object satisfies the condition in its scope, and then to move on to the next object. It is obvious that one can only evaluate the egg that one just holding in one's hand. Similarly, the quantifier can only evaluate the object it is considering at the moment. In order not to mix up the eggs/objects, (50) used the expressions 'this object', which serves the purpose of keeping control of which object is currently evaluates. The expression 'this object' keeps constant the particular object temporarily picked, and it does so till the computation reaches the end of the scope of the quantifier. Since‘this object’ may refer to many different actual objects (many eggs), ‘this object’ is said to serve as a ***variable***.

In the example at hand, which involves the quantifier *every,* the whole sentence is evaluated as true if all possible ways of picking an object from the available ones leads to a true statement.

● To illustrate how formulas are evaluated in specific situations, consider (51)a and its interpretation in the three scenarios below.

(51) a. Every bullet is black
b. For every thing: if this thing is a bullet, then this thing is also a black

(52) Scenario I:

a.	$●_1$	$●_1$	$●_1$	TRUE
b.	$●_2$	$●_2$	$●_2$	TRUE

➜ Sentence is true in Scenario I

(53) Scenario II:

a.	$●_1$	$●_1$	$●_1$	TRUE
b.	$●_2$	$●_2$	$●_2$	TRUE
c.	$●_3$	$●_3$	$○_3$	FALSE

➜ Sentence is false in Scenario II

(54) Scenario III:

a.	$●_1$	$●_1$	$●_1$	TRUE
b.	$●_2$	$○_2$	$●_2$	TRUE
c.	$●_3$	$●_3$	$●_3$	TRUE

➜ Sentence is true in Scenario III

In order not to be forced to always use a cumbersome natural language paraphrase referring to ‘the thing’ under consideration, the expression is substituted by a (shorter) variable, usually x,y,z,... for individuals. Moreover, for the expression ‘for all’, we use the symbol $\forall$. This yields the following translation intro quantified predicate logic for (51):

(55) $\forall$x[bullet(x) $\rightarrow$ black(x)]

2.3.2. Existential Statements

In a similar way, it is possible to translate sentences with an existential quantifier such as *one, at least one, some,* or *a(n)*. Here, we use the symbold ‘$\exists$’, the so-called existential quantifier, and the connective ‘$\wedge$’ instead of ‘$\rightarrow$’:

(56) a. Some/A/One bullet is black
b. $\exists$y[buller(y) $\wedge$ black(y)]

(57) *Question*: Can (56) be translated by the use of ‘$\rightarrow$’? If not, why?

The following examples illustrate how to translate slightly more complex sentences into predicate logic, highlighting common sources for mistakes.

2.3.3. Quantifiers, coordination and Scope

● Potential source of confusion: If a universal quantifier has scope over a disjunction (a coordination involving *or*), the correct bracketing is important. In some cases, as in (58), adding brackets at the wrong place (see (59)b) yields an incorrect result. (59)b would e.g. be true if the world only consisted of white objects, while (58) makes a restricted claim about wine only:

(58) Every wine is red or white

(59) a. $\forall x[\text{wine}(x) \rightarrow [\text{red}(x) \vee \text{white}(x)]]$

b. $\forall x[[\text{wine}(x) \rightarrow \text{red}(x)] \vee \text{white}(x)] =$
$\forall x[\text{white}(x) \vee [\text{wine}(x) \rightarrow \text{red}(x)]]$
"Everything is white, or - if it is wine - red"

● Similarly, it is important to pay attention whenever a conjunction (coordination with *and*) is combined with an indefinite, i.e. NPs introduced by *a(n)* or other NPs which are standardly translated as universal quantifiers (*some NP, one NP, at least one NP*), as shown by (60) and (61):

(60) a. A man came and left
b. $\exists x[\text{man}(x) \wedge \text{came}(x) \wedge \text{left}(x)]$

(61) a. A man came and a man left
b. $\exists x[\text{man}(x) \wedge \text{came}(x)]_{\text{End of scope of }\exists} \wedge \exists x[\text{man}(x) \wedge \text{left}(x)]_{\text{End of scope of }\exists}$

(60)a denotes the proposition that a man came and *the same man who came* left. This 'sameness' is expressed by using one quantifier (∃) which binds a variable in the argument position of *came* as well as in the argument position of *left.* This ensures that the sentence is true whenever 'there is *an x*, such that *x* is a man and *x* came and *x* left'. In (61)a, on the other hand, the individuals who arrived and left can - but don't have to - be represented by two different men. The fact that there are potentially two different men involved in the two actions is captured by the formula in (61)b, which uses two existential quantifiers. Crucially, the scope of the first quantifier extends only to the end of the first closing bracket (as indicated by the subscript). Since quantifiers only bind variables which are inside their scope, '∃x' only binds all occurrences of x up to the first occurrence of ']'. Given that the scope of the first '∃x' ends before the conjunction ('∧'), it also follows that the second occurrence of ∃x is interpreted as if the first one would not exist. For this reason, one can write '∃x', using the same variable 'x' again. Note however that it would equally be possible (and maybe less confusing) to use another variable, say 'z'. (62) and (61)b are accordingly equivalent.

(62) $\exists x[\text{man}(x) \wedge \text{came}(x)]_{\text{End of scope of }\exists} \wedge \exists z[\text{man}(z) \wedge \text{left}(z)]_{\text{End of scope of }\exists}$

○ Further illustration: In (63), the quantifier cannot bind the higher occurence of the variable z (in 'sleep(z)'), because it does not take scope over that variable:

(63) $\text{sleep}(z) \wedge \forall z[\text{dog}(z) \rightarrow \text{smell}(z)]$

○ In (64), the variable x which *boy* predicates over is only bound by the lower operator, which is closer than the higher existential, because x is not free at the level designated by ☛

(64) $\exists x$ ☛$[\exists x[boy(x) \wedge read(x)] \wedge sleep(x)]$ = *lower, local ∃ binds first two x's*
= $\exists x[boy(x) \wedge read(x)] \wedge \exists x[sleep(x)]$ ≠
≠ $\exists x[boy(x) \wedge read(x) \wedge sleep(x)]$

2.3.4. Interdefinability: Equivalences with ∃ and ∀

If a formula involves one type of quantifier (∃ or ∀) and negation, it can be translated into another, equivalent formula which uses the other type of quantifier (i.e. ∀ instead of ∃) and locates the negation in a different, yet systematically related position.

(65) *Observation*: The following two statements are equivalent:

(66) a. No animal bears fruit
$\neg\exists x[animal(x) \wedge bears\_fruit(x)]$
b. All animals (are such that they) don't bear fruit
$\forall x[animal(x) \rightarrow \neg bears\_fruit(x)]$

More generally, the following equivalence holds:

(67) A formula in which negation has scope over an existential quantifier (*no NP)* is equivalent to a formula with a universal quantifier in which negation takes scope below the connective:

$$\neg\exists x[A(x) \wedge B(x)] \quad \Leftrightarrow \quad \forall x[A(x) \rightarrow \neg B(x)]$$

(68) *Observation*: The following two statements are equivalent:

(69) a. Not every plant bears fruit
$\neg\forall x[plant(x) \rightarrow bears\_fruit(x)]$
b. At least one plant does not bear fruit
$\exists x[plant(x) \wedge \neg bears\_fruit(x)]$

More generally, the following equivalence holds (it is the same as in (67), the only difference being that *universal* has been replaced by *existential* and v.v.)

(70) A formula in which negation has scope over a universal quantifier (*not every NP*) is equivalent to a formula with an existential quantifier in which negation takes scope below the connective:

$$\neg\forall x[A(x) \rightarrow B(x)] \quad \Leftrightarrow \quad \exists x[A(x) \wedge \neg B(x)]$$

2.3.5. Brief summary predicate logic

(71)	*Quantifier*	*Paraphrase*	*Example*
	a. $\exists x$	"there is an x such that"	$\exists x[\text{planet}(x) \wedge \text{blue}(x)]$ "There is an x such that x is a planet and x is blue"
	b. $\forall x$	"for every x, if"	$\forall x[\text{planet}(x) \rightarrow \text{round}(x)]$ "For every x, if x is a planet then x is round"
	c. $\neg\exists x$	"there is no x such that"	$\neg\exists x[\text{planet}(x) \wedge \text{square}(x)]$ "There is no x such that x is a planet and x is square"
	d. $\neg\forall x$	"for not every x, if"	$\neg\forall x[\text{planet}(x) \rightarrow \text{blue}(x)]$ "For not every x, if x is a planet then x is blue"

● How to formulate equivalences between formulas involving $\exists$ and $\forall$, respectively:
(i) change quantifier (from $\exists$ to $\forall$ and from $\forall$ to $\exists$, marked ⥣) and
(ii) change position of negation ($\neg$, marked by ▲)

(72)	*Equivalence*	*Example*
	a. $\neg\exists x[A(x) \wedge B(x)] \Leftrightarrow \forall x[A(x) \wedge \neg B(x)]$ ▲⥣ ⥣ ▲	$\neg\exists x[\text{planet}(x) \wedge \text{square}(x)] \Leftrightarrow$ $\Leftrightarrow \forall x[\text{planet}(x) \rightarrow \neg\text{square}(x)]$
	b. $\neg\forall x[A(x) \rightarrow B(x)] \Leftrightarrow \exists x[A(x) \wedge \neg B(x)]$ ▲⥣ ⥣ ▲	$\neg\forall x[\text{planet}(x) \rightarrow \text{blue}(x)] \Leftrightarrow$ $\Leftrightarrow \exists x[\text{planet}(x) \wedge \neg\text{blue}(x)]$

<u>Exercises</u>

● Translate the following *unambiguous* sentences into predicate logic. Use upper case letters for predicates and lower case letters for arguments (names, variables).

(73) All girls arrived early
(74) He invited each friend
(75) Nobody called Fred
(76) Sally bought something interesting
(77) We gave Bill every cent we had (NB: *give* is a three-place predicate!)
(78) You owe them nothing

● Translate the following *ambiguous* sentences into predicate logic. Since they are two-way ambiguous, you'll need two different translations for each sentence:

(79) A girl climbed every tree
(80) They didn't see one movie
(81) Everybody didn't like Sam
(82) Every critic liked all the paintings
(83) One player didn't win

2.3.6. Application: Entailment/Monotonicity & Inferences in NLS

One of the many applications of quantificational predicate logic in NL analysis pertains to semantic impact a particular choice of quantifier has on the meaning of the rest of the sentence. Studies of these properties make it e.g. possible to formulate systematic correlations between purely linguistic phenomena such as NPI licensing and inference patterns.

Semantics is concerned with *meaning relations,* not meanings themselves. This property is clearly foregrounded in the calculation of *inferences*, which can be reduced to syntactic schemata such as the syllologisms of classical (Aristotelian and scholastic) logic. One prominent inference rule is Existence Generalization:

(84) *Existence Generalization (EG)*

a. Wood floates → Something floats
b. Schema: NP VP → ∃xVP

But EG does not hold for all contexts. Its validity crucially depends on the kind of argumens the predicate is combined with:

(85) Nothing floats =//=> Something floats

Whether EG holds or not can be read off an independent properties of the meanings of the parts of the sentence (NP and VP). In particular, the validity of EG is related to *monotonicity or entailment* properties of the NP and the VP.

● Another prominent semantic property which can (at least partially) be explained in terms of *entailment* is the behavior of Negative Polarity Items (NPIs - examples repeated from above):

(86) a. Sam hasn't ever heard about it
b. Nobody has ever heard about it.
c. Everybody who has ever heard about it likes it
d. Many men/at most two dozen people have ever heard about it

(87) a. *Sam has ever heard about it
b. *Everybody has ever heard about it.
c. *Somebody who has ever heard about it likes it
d. *Few men/at least two dozen people have ever heard about it

<u>2.3.6.1. Upward Entailment</u>

Functions can be categorized according to their entailment properties. Some functions are upward entailing (or monotone increasing):

(88) A function f is *upward entailing* iff for any X, Y in the domain of f,
if $X \subseteq Y$ then $f(X) \subseteq f(Y)$

○ For instance, the set of individuals which *arrived yesterday morning* is subset of individuals who *arrived yesterday*:

(89) a. ⟦arrived yesterday morning⟧ = X
⟦arrived yesterday⟧ = Y

b. ⟦arrived yesterday morning⟧ ⊆ $X \subseteq Y$
⟦arrived yesterday⟧

○ *all* and *some* are upward entailing (let f be **all boys** or **some boys**):

(90) a. All/some boys arrived yesterday
b. All/some boys arrived yesterday morning ⇑

2.3.6.2. Downward Entailment

(91) A function f is *downward entailing* iff for any X, Y in the domain of f,
if $X \subseteq Y$ then $f(Y) \subseteq f(X)$

○ *Nobody* is e.g. a downward entailing (or monotone decreasing; ⇓) function:

(92) ⟦arrived yesterday morning⟧ = X
⟦arrived yesterday⟧ = Y

(93) a. Nobody arrived yesterday ⇓
b. Nobody arrived yesterday morning

(94) ⟦Nobody arrived yesterday⟧ ⊆ $Y \subseteq X$
⟦nobody arrived yesterday morning⟧

○ *few men* is a monotone decreasing function:

(95) a. Few men arrived yesterday ⇓
b. Few men arrived yesterday morning

(96) ⟦few men arrived yesterday⟧ ⊆
⟦few men arrived yesterday morning⟧

○ *at most two dozen people* creates downward entailing contexts:

(97) a. At most two dozen people arrived yesterday ⇓
b. At most two dozen people arrived yesterday morning

(98) ⟦at most two dozen people arrived yesterday⟧ ⊆
⟦at most two dozen people arrived yesterday morning⟧

(99) *Ladusaw's Generalization*

Only downward entailing contexts license Negative Polarity Items (NPI):

(100) a. Sam hasn't ever heard about it
b. Nobody has ever heard about it
c. Few people have ever heard about it
d. At most two dozen people have ever heard about it

(101) a. *Sam has ever heard about it
b. *Everybody has ever heard about it
c. *Many people have ever heard about it
d. *At least two dozen people have ever heard about it

(102) *Question*: Does Ladusaw's generalization extend to the following examples?

(103) a. Everybody who has ever read it likes it
b. Have you ever read it?

(104) a. Everybody who has read it likes it
b. Everybody who has read it in school likes it

(105) a. Have you read it?
b. Have you read it in school?

Exercise

In (106) *some movie* has to take scope over negation, while *a movie* doesn't have to do so. (i) Provide a scenario which can be described by (106)a, but not by (106)b. (ii) Why is there this imbalance?

(106) a. John didn't see some movie.
b. John didn't see a movie.

2.3.7. Formalization of Quantificational Predicate Logic

Quantificational predicate logic, just like predicate calculus, is a formal language. Formal languages consist of (i) a vocabulary or lexicon, (ii) a syntax, and (iii) a semantics.

(107) *Vocabulary/Lexicon:*

a. Terms: Individual constants: *sally, john,...*
Individual variables: *x, y, z, ...*

b. Predicates: *play, sleep, hit, eat, give,...*

c. Logical connectives:

$\wedge$	Conjunction ("and")
$\vee$	Disjunction ("or"; from Latin *vel*)
$\neg$	Negation ("not"; also ~)
$\rightarrow$	Material Implication ("if ... then")
$\leftrightarrow$	Biconditional (*iff* "if and only if")

d. Quantifiers: $\exists$, $\forall$

e. Brackets: [,]

(108) *Semantics:*

a. If P is an n-place relation, and t_1,.... t_i are terms, then P(t_1, ...t_i) is a *formula*

b. If φ and ψ are formulas, then

i. $[\![\neg\varphi]\!]$ = 1 iff $[\![\varphi]\!] = 0$

ii. $[\![\varphi \wedge \psi]\!]$ = 1 iff $[\![\varphi]\!] = [\![\psi]\!] = 1$

iii. $[\![\varphi \vee \psi]\!]$ = 1 iff $[\![\varphi]\!] = 1$ or $[\![\psi]\!] = 1$

iv. $[\![\varphi \rightarrow \psi]\!]$ = 1 iff $[\![\varphi]\!] = 0$ or $[\![\psi]\!] = 1$

v. $[\![\varphi \leftrightarrow \psi]\!]$ = 1 iff $[\![\varphi]\!] = [\![\psi]\!]$

c. i. $[\![\exists x\varphi]\!]$ = 1 iff substituting x in φ by some a which is an x yields 1

ii. $[\![\forall x\varphi]\!]$ = 1 iff substituting x in φ by any a which is an x yields 1

NB: The simplified version of the semantics provided above is included only for complementes sake. The actual semantics, in particular that of ∃ and ∀ is somewhat more complex and involves what is called *temporal variable assignments*. For an introduction see e.g. GAMUT I, p. 87.

Some web resources on Logic (for updates see also course webpage)

- Excellent logic webpage with good interactive exercises, starts from scratch:
  http://people.hofstra.edu/faculty/Stefan_Waner/RealWorld/logic/logic1.html
- How to read logical notation:
  http://www.philosophy.ed.ac.uk/study_html/vade_mecum/sections/section3/3-1.htm
- For those with some background: http://www.rbjones.com/rbjpub/logic/
- Glossary of first order logic: http://www.rbjones.com/rbjpub/logic/log004.htm

NEXT:
- Implementing a compositional semantics of a fragment of English
- Determining the semantic composition rules
- Extending formal tool kit (λ-calculus, modified variable assignment)

3. Intransitive Constructions

(Sections below follow Heim & Kratzer 1998)

Goal: Compositional derivation of the denotation of (1) (= its truth-conditions) from the meaning of its parts.

(1) Bart smokes.

3.1. Function Application

Consider the fine-grained structure of (1), and the isomorph (= same_structure) examples in (3):

(2) $\llbracket[_{IP}[_{NP}[_{N°}\ \textbf{Bart}]]\ [_{VP}\ [_{V°}\ \textbf{smokes}]]]\rrbracket$ = 1 iff Bart smokes

(3) a. $\llbracket[_{IP}[_{NP}[_{N°}\ \textbf{Joe}]\ [_{VP}\ [_{V°}\ \textbf{arrived}]]]\rrbracket$ = 1 iff Joe arrived
b. $\llbracket[_{IP}[_{NP}[_{N°}\ \textbf{Sam}]\ [_{VP}\ [_{V°}\ \textbf{slept}]]]\rrbracket$ = 1 iff Sam slept

Whatever constituent is substituted for the subject NP or the VP, the principle of composition remains the same:

(4) *Corollary*: For any sentence of the shape subject^predicate, it holds that
(i) it can be *syntactically* decomposed into a subject and a predicate and
(ii) it can be *semantically* interpreted in such a way that the syntactic subject is the object of the function denoted by the predicate.

● Note that the syntactic *subject* serves as a semantic *object* of the function/predicates!

<u>Fragment E1</u>

A. *Denotations*

α is a possible denotation of $\llbracket\alpha\rrbracket$ iff α is either an element of (i), (ii) or the set in (iii):
(i) D: set of individuals
(ii) {0,1}: set of truth values
(iii) Functions from D to {0,1}

B. *Lexicon*: Semantic translation rules for terminals:

$\llbracket$**Jeff**$\rrbracket$	=	Jeff
$\llbracket$**Bart**$\rrbracket$	=	Bart
$\llbracket$**smokes**$\rrbracket$	=	f: D → {0,1}, such that for any a∈D, f(a) = 1 iff a smokes
$\llbracket$**sleeps**$\rrbracket$	=	f: D → {0,1}, s.t. for any a∈D, f(a) = 1 iff a sleeps

C. *Semantic translation rules for non-terminals:*

S1. Sentence Rule: If α has the form $[_{IP}\ \beta\ \gamma]$, then $\llbracket\alpha\rrbracket = \llbracket\gamma\rrbracket(\llbracket\beta\rrbracket)$ (Function Application)

S2. Non-branching nodes: If α has the form $[_{Xn}\ \beta]$, where n∈{°,’,P}, then $\llbracket\alpha\rrbracket = \llbracket\beta\rrbracket$

Sample Derivation:

(5)

1. $\llbracket [_{IP} [_{NP} [_{N^\circ} \textbf{Bart}]] [_{VP} [_{V^\circ} \textbf{smokes}]]] \rrbracket \;=\; \llbracket [_{VP} [_{V^\circ} \textbf{smokes}]] \rrbracket (\llbracket [_{NP} [_{N^\circ} \textbf{Bart}]] \rrbracket)$ *S1*
2. $\llbracket [_{VP} [_{V^\circ} \textbf{smokes}]] \rrbracket \;=\; \llbracket [_{V^\circ} \textbf{smokes}] \rrbracket \;=\; \llbracket \textbf{smokes} \rrbracket$ *2 x S2*
3. $\llbracket [_{NP} [_{N^\circ} \textbf{Bart}]] \rrbracket \;=\; \llbracket [_{N^\circ} \textbf{Bart}] \rrbracket \;=\; \llbracket \textbf{Bart} \rrbracket$ *2 x S2*
4. $\llbracket [_{IP} [_{NP} [_{N^\circ} \textbf{Bart}]] [_{VP} [_{V^\circ} \textbf{smokes}]]] \rrbracket \;=\; \llbracket \textbf{smokes} \rrbracket (\llbracket \textbf{Bart} \rrbracket)$ *from 1. by substitution*
5. $\llbracket \textbf{smokes} \rrbracket \;=\; \begin{bmatrix} \text{f: D} \rightarrow \{0,1\} \text{ s.t. for any a} \in \text{D,} \\ \text{f(a) = 1 iff a smokes} \end{bmatrix}$ *Lexicon*
6. $\llbracket \textbf{Bart} \rrbracket \;=\; \text{Bart}$ *Lexicon*

   from 4., by lexical insertion
7. $\llbracket \textbf{smokes} \rrbracket (\llbracket \textbf{Bart} \rrbracket) \;=\; \begin{bmatrix} \text{f: D} \rightarrow \{0,1\} \text{ s.t. for any a} \in \text{D,} \\ \text{f(a) = 1 iff a smokes} \end{bmatrix} \text{(Bart)}$

   Function application: f applies to 'Bart'
8. $\begin{bmatrix} \text{f: D} \rightarrow \{0,1\} \text{ s.t. for any a} \in \text{D,} \\ \text{f(a) = 1 iff a smokes} \end{bmatrix} \text{(Bart)} \;=\; \text{f(Bart) = 1 iff Bart smokes}$

   from 8. by substitution
9. $\llbracket [_{IP} [_{NP} [_{N^\circ} \textbf{Bart}]] [_{VP} [_{V^\circ} \textbf{smokes}]]] \rrbracket \;=\; \text{1 iff Bart smokes}$

QED

Step 8, which depicts function application, is the central one in the computation. In what follows, the procedure will be spelled out in more detail.

Reduction: stepwise substitution ('rewriting') of an expression α by an expression β according to some set of rewriting rules. In the example below, rewriting results in evaluation of parts of the expression:

(6)
$$\begin{aligned} (3+7) \text{ x } (8/2 - 1) &= 10 \text{ x } (4 - 1) = \\ &= 10 \text{ x } 3 = \\ &= 30 \end{aligned}$$

Function application: leads to a reduction of the arity of the relation (i.e. the number of arguments it takes). Given that the operation is defined for 'any a...' of a suitable type, and given that the terms '2' and 'Bart' are suitable a's (they are members of ℕ and D, respectively, the variable x can be instantiated by these terms. Once the function is applied to one of its arguments, it returns as an output the value on the right-hand side of the equation:

(7)

a. $\begin{bmatrix} \text{f: } \mathbb{N} \rightarrow \mathbb{N} \text{ s.t. for any a} \in \mathbb{N} \\ \text{f(x)} \quad = \quad x^2 \end{bmatrix} (2) \quad \Rightarrow \quad 2^2 \; (= 4)$

b. $\begin{bmatrix} \text{f: D} \rightarrow \text{D s.t. for any a} \in \text{D} \\ \underbrace{\text{f(x)}}_{\text{Function plus Variable}} \quad = \quad \underbrace{\text{1 iff x smokes}}_{\text{Value of function}} \end{bmatrix} \underbrace{\text{(Bart)}}_{\text{Argument}} \quad \underset{\text{FA}}{\Rightarrow} \quad \underbrace{\text{1 iff Bart smokes}}_{\text{Result of Function Application}}$

● The result of Function Application can then be substituted in the derivation, yielding the equivalence in step 9. More precisely, the substituions are justified in the following way:

Function Application reduces the part inside the larger box (8)b to the part inside the box in (8)c. Furthermore, since the sentence denotation is equivalent to the boxed part of (8)b (by (8)a), and the boxed part of (8)b is equivalent to the boxed part in (8)c, it follows that the sentence denotation is equivalent to the boxed part in (8)d. It is exactly equivalence which underlies the crucial step in the derivation, expressed by step 9.

(8) a. $[\![[_{IP} [_{NP} [_{N°}$ **Bart**]] $[_{VP} [_{V°}$ **smokes**]]] $]\!]$ = $[\![$**smokes**$]\!]([\![$**Bart**$]\!])$

b. $[\![$**smokes**$]\!]([\![$**Bart**$]\!])$ = $\boxed{\begin{bmatrix} \text{f: D} \rightarrow \{0,1\} \text{ s.t. for any a} \in \text{D,} \\ \text{f(a) = } \boxed{\text{1 iff a smokes}} \end{bmatrix} \text{(Bart)}}$

⇓ ⇓
Function Application
⇓ ⇓

c. $[\![$**smokes**$]\!]([\![$**Bart**$]\!])$ = $\boxed{\text{1 iff Bart smokes}}$

d. $[\![[_{IP} [_{NP} [_{N°}$ **Bart**]] $[_{VP} [_{V°}$ **smokes**]]] $]\!]$ = $\boxed{\text{1 iff Bart smokes}}$ (= Step 9)

3.2. MODEL THEORY

Constant symbols: somewhat idealized, names like *Jeff* and *Bart* denote the same individual in all situations: they are *rigid designators.* But *definite descriptions* such as *the president* may not be treated in the same way, as the interpretation function returns different individuals (or the concept thereof), depending on which president one refers to (of a company, of a club, of the USA,...)

(9) a. $[\![$**Bart**$]\!]$ = Bart
b. $[\![$**the_president$_1$**$]\!]$ = George W. Bush
$[\![$**the_president$_2$**$]\!]$ = Sally Smith (the president of GM)
.....

Moreover, what about lexically ambiguous expression such as e.g. [katse], which means 'sit!' in Greek, but 'cat' in German? By definition, a function maps an object from its domain to one and only one object within its range. It follows that the interpretation function $[\![\cdot]\!]$ cannot map [katse] to its appropriate meanings in both languages. That is, one member of the pair of homophones would end up without an interpretation, counter to the intuitions of native speakers of Greek and German, respectively.

● The standard solution consists in interpreting expressions relative to a *model*, which assigns meanings to natural language expressions. Sentences and their components are mapped on to the elements of a model M by an interpretation function I (from Weiss & d'Mello 1997):

(10) A Model for a Language L is an ordered pair <A,I>, where A is a set (the universe) and I is an interpretation function whose domain is the set of all constants, relations and function symbols of L such that

- (i) If c is a constant symbol, then I(c) $\in$ A
- (ii) If F is a function symbol, then I(F) is a function on A
- (iii) If R is an m-place relation symbol, then I(R) is an m-place relation on A

More generally, models makes it possible to study the properties of formal systems without being forced to consider all possible extensions of every single symbol.

Example

The real numbers, the ordering relation (FOLLOW) and the arithmetic operations of multiplication (MULTIPLY) and addition can be conceived of as a model. A fragment might look as follows:

(11) a. M = <A, I>
A = set of real numbers
b. I(FOLLOW) = <
I(MULTIPLY) = •

(12) a. For every number, there is a number which FOLLOWS it
b. nine MULTIPLIED with nine

● On this view, expressions in different languages are interpreted differently because they are part of different models. Returning to the original puzzle of homophony, a simple model theoretic reconstruction yields the following:

(13) a. Model M-Greek = <A-Greek, I-Greek>
b. A-Greek = {[katse], gata, skillos, alepu, potitki, skiouros, ...}
c. I-Greek(**katse**) = 'sit'

(14) a. Model M-Germ = <A-Germ, I-Germ>
b. A-Greek = {sitz, [katse], hund, fuchs, maus, eichhörnchen, ...}
c. I-Germ(**katse**) = 'cat'

● In formal semantics, interpretation relative to a model is usually indicated by superscripting the model (M) to the denotation brackets:

(15) $[\![\Phi]\!]^{M}$ = is the function which yields the interpretation of Φ relative to M

(16) a. $[\![\textbf{katse}]\!]^{\text{M-Greek}}$ = I-Greek (katse) = 'sit'
b. $[\![\textbf{katse}]\!]^{\text{M-Germ}}$ = I-Germ (katse) = 'cat'

NOTATIONAL CONVENTION: Unless required, superscripts will be suppressed.

Resume

- The meaning of a sentence is defined by its truth-conditions.
- NL semantics is computed compositionally (this is a heuristics, not a fact!)
- Semantics operates on syntactic trees.
- FA derives the meaning of branching nodes from the meaning of its daughters.

3.3. EXTENSIONS AND EVALUATION

The fragment E1 provides an algorithm to assign truth conditions to simple intransitive sentences of English. However, it is not know yet whether the sentences are true or false in a specific scenario. For instance, if Bart and Jeff are the only individuals in a world, and if in that world Jeff smokes and Bart sleeps, we intuitively know that sentence (2) comes out as false. In order to arrive at this result, it is necessary to consider the *extension* of the predicates in an actual scenario, and not just the extensions of their lexical entries as was done above. Extensions are the denotations of expressions in an actual situation or world.

Above, predicates were taken to denote functions. Moreover, functions are a special kind of relations, which can be defined as a set of ordered pairs. One way to describe the extensions of the predicates in the scenario above consists in listing the ordered pairs as a set or in a table:

(17) EXTENSIONS IN SCENARIO SC1:

$[\![\textbf{smoke}]\!] = \{\langle \text{Bart}, 0\rangle, \langle \text{Jeff}, 1\rangle\} =$ *Set notation*

$= \begin{bmatrix} \text{Bart} & \rightarrow & 0 \\ \text{Jeff} & \rightarrow & 1 \end{bmatrix}$ *Function notation* (*table*)

$[\![\textbf{sleep}]\!] = \{\langle \text{Bart}, 1\rangle, \langle \text{Jeff}, 1\rangle\} =$

$= \begin{bmatrix} \text{Bart} & \rightarrow & 1 \\ \text{Jeff} & \rightarrow & 1 \end{bmatrix}$

Then, the denotation of sentence (2) in scenario SC1 can be given as below:

(18)
1. $[\![[_{IP}[_{NP}[_{N^\circ}\ \textbf{Bart}]]\ [_{VP}\ [_{V^\circ}\ \textbf{smokes}]]]]\!] = [\![[_{VP}\ [_{V^\circ}\ \textbf{smokes}]]]\!]([\![\ [_{NP}[_{N^\circ}\ \textbf{Bart}]]]\!])$ *S1*
2. $[\![[_{VP}\ [_{V^\circ}\ \textbf{smokes}]]]\!] = [\![[_{V^\circ}\ \textbf{smokes}]]\!] = [\![\textbf{smokes}]\!]$ *2 x S2*
3. $[\![\ [_{NP}[_{N^\circ}\ \textbf{Bart}]]]\!] = [\![[_{N^\circ}\ \textbf{Bart}]]\!] = [\![\textbf{Bart}]\!]$ *2 x S2*
4. $[\![[_{IP}[_{NP}[_{N^\circ}\ \textbf{Bart}]]\ [_{VP}\ [_{V^\circ}\ \textbf{smokes}]]]]\!] = [\![\textbf{smokes}]\!]([\![\textbf{Bart}]\!])$ *from 1. by substitution*
5. $[\![\textbf{smokes}]\!] = \begin{bmatrix} \text{Bart} \rightarrow 0 \\ \text{Jeff} \rightarrow 1 \end{bmatrix}$ *Extension in SC1*
6. $[\![\textbf{Bart}]\!] = \text{Bart}$ *Extension in SC1*
7. $[\![\textbf{smokes}]\!]([\![\textbf{Bart}]\!]) = \begin{bmatrix} \text{Bart} \rightarrow 0 \\ \text{Jeff} \rightarrow 1 \end{bmatrix}(\text{Bart}) = 0$ *from 4. by substitution*

In (18), the denotation of the predicate is not its lexical entry, but its extension in a scenario, and (2) can now be evaluated since (2) is paired with a truth value, and no longer with its truth-conditions: Applying the function $[\![\textbf{smoke}]\!]$ to $[\![\textbf{Bart}]\!]$ yields 0, and the sentence correctly comes out as false in the scenario. Functions such as in (17) which range over truth values are also called *characteristic functions*, because they characterize whether an element of the domain has a certain property or not (see below).

Bottom-up computation

Above, (2) was proved in a top-down fashion, such that the computation started with the denotation of the whole clause and decomposed the clause in a step-by-step fashion. But it is equally possible to prove (2) from bottom-up, both strategies yield the same results:

(2) $\llbracket [_{IP} [_{NP} [_{N^\circ} \textbf{Bart}]] [_{VP} [_{V^\circ} \textbf{smokes}]]] \rrbracket$ = 1 iff Bart smokes

(19)

1. $\llbracket \textbf{Bart} \rrbracket$ = Bart *Lexicon*
2. $\llbracket \textbf{smokes} \rrbracket = \begin{bmatrix} f: D \rightarrow \{0,1\} \text{ s.t. for any } a \in D, \\ f(a) = 1 \text{ iff } a \text{ smokes} \end{bmatrix}$ *Lexicon*
3. $\llbracket \textbf{Bart} \rrbracket = \llbracket [_{N^\circ} \textbf{Bart}] \rrbracket = \llbracket [_{NP} [_{N^\circ} \textbf{Bart}]] \rrbracket$ *2 x S2*
4. $\llbracket \textbf{smokes} \rrbracket = \llbracket [_{V^\circ} \textbf{smokes}] \rrbracket = \llbracket [_{VP} [_{V^\circ} \textbf{smokes}]] \rrbracket$ *2 x S2*
5. $\llbracket [_{IP} [_{NP} [_{N^\circ} \textbf{Bart}]] [_{VP} [_{V^\circ} \textbf{smokes}]]] \rrbracket = \llbracket [_{VP} [_{V^\circ} \textbf{smokes}]] \rrbracket (\llbracket [_{NP} [_{N^\circ} \textbf{Bart}]] \rrbracket)$ *S1*
6. $\llbracket [_{IP} [_{NP} [_{N^\circ} \textbf{Bart}]] [_{VP} [_{V^\circ} \textbf{smokes}]]] \rrbracket = \llbracket \textbf{smokes} \rrbracket (\llbracket \textbf{Bart} \rrbracket)$ *from 5. by substitution*
7. $\llbracket \textbf{smokes} \rrbracket (\llbracket \textbf{Bart} \rrbracket) = \begin{bmatrix} f: D \rightarrow \{0,1\} \text{ s.t. for any } a \in D, \\ f(a) = 1 \text{ iff } a \text{ smokes} \end{bmatrix} (\text{Bart})$ *from 4. by lexical insertion*
8. $\begin{bmatrix} f: D \rightarrow \{0,1\} \text{ s.t. for any } a \in D, \\ f(a) = 1 \text{ iff } a \text{ smokes} \end{bmatrix} (\text{Bart}) = f(\text{Bart}) = 1 \text{ iff Bart smokes}$ *applying f to 'Bart'*
9. $\llbracket [_{IP} [_{NP} [_{N^\circ} \textbf{Bart}]] [_{VP} [_{V^\circ} \textbf{smokes}]]] \rrbracket$ = 1 iff Bart smokes *from 8. by substitution*

Application: Connectives

With this much in the background, one can design lexical entries and translation rules for the connectives such as **and** and **or** which accounts for their occurrence in contexts such as:

(20) a. Bart or Jeff smoke
b. Jeff smokes or sleeps

Even though the analysis is rather awkward - and, as it will turn, ultimately untenable - let us consider an explicit version of the semantics of disjunction as an exercise in formulating rules and computing them:

Fragment E2

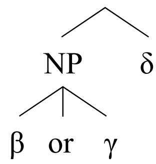
Lexicon

$\llbracket \mathbf{or_1} \rrbracket$ = f: $D \to (D \to ((D \to \{0,1\}) \to \{0,1\}))$ s. t. for any $a,b \in D$ and g: $D \to \{0,1\}$,
f(a)(b)(g) = 1 iff g(a) = 1 or g(b) = 1

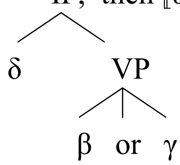
$\llbracket \mathbf{or_2} \rrbracket$ = f: $(D \to \{0,1\}) \to ((D \to \{0,1\}) \to (D \to \{0,1\}))$ s. t. for any $a \in D$ and g,h: $D \to \{0,1\}$,
f(g)(h)(a) = 1 iff g(a) = 1 or h(a) = 1

Semantic translation rules

D1. ***NP-Disjunction:*** If α has the form IP, then $\llbracket \alpha \rrbracket = \llbracket \mathbf{or_1} \rrbracket(\llbracket \beta \rrbracket)(\llbracket \gamma \rrbracket)(\llbracket \delta \rrbracket)$



D2. ***VP-Disjunction:*** If α has the form IP, then $\llbracket \alpha \rrbracket = \llbracket \mathbf{or_2} \rrbracket(\llbracket \beta \rrbracket)(\llbracket \gamma \rrbracket)(\llbracket \delta \rrbracket)$



Sample Computation: (20)a

(21)

1. $\llbracket [_{IP}$ **Bart or Jeff smoke**$] \rrbracket$ = $\llbracket \mathbf{or_1} \rrbracket(\llbracket [_{NP}$ **Bart**$] \rrbracket)(\llbracket [_{NP}$ **Jeff**$] \rrbracket)(\llbracket [_{VP}$ **smoke**$] \rrbracket)$ *D1*
2. $\llbracket [_{NP}$ **Bart**$] \rrbracket$ = $\llbracket$**Bart**$\rrbracket$ *S2*
3. $\llbracket [_{NP}$ **Jeff**$] \rrbracket$ = $\llbracket$**Jeff**$\rrbracket$ *S2*
4. $\llbracket [_{VP}$ **smoke**$] \rrbracket$ = $\llbracket$**smoke**$\rrbracket$
5. $\llbracket [_{IP}$ **Bart or Jeff smoke**$] \rrbracket$ = $\llbracket \mathbf{or_1} \rrbracket(\llbracket$**Bart**$\rrbracket)(\llbracket$**Jeff**$\rrbracket)(\llbracket$**smoke**$\rrbracket)$ *substitution*
6. $\llbracket \mathbf{or_1} \rrbracket(\llbracket$**Bart**$\rrbracket)(\llbracket$**Jeff**$\rrbracket)(\llbracket$**smoke**$\rrbracket)$ = *Lexicon*

$$= \left[\begin{array}{l} \text{f: } D \to (D \to ((D \to \{0,1\}) \to \{0,1\})) \text{ s.t. for any } a,b \in D \\ \text{and g: } D \to \{0,1\}, \text{f(a)(b)(g)} = 1 \text{ iff g(a)} = 1 \text{ or g(b)} = 1 \end{array}\right](\text{Bart})(\text{Jeff})(\text{smoke})$$

7. $$\left[\begin{array}{l} \text{f: } D \to (D \to ((D \to \{0,1\}) \to \{0,1\})) \text{ s.t. for any } a,b \in D \\ \text{and g: } D \to \{0,1\}, \text{f(a)(b)(g)} = 1 \text{ iff g(a)} = 1 \text{ or g(b)} = 1 \end{array}\right](\text{Bart})(\text{Jeff})(\text{smoke}) = 1$$

iff Bart smokes or Jeff smokes

Sample Computation: (20)b

(22) *D2*

1. $[\![[_{IP}$ **Jeff smokes or sleeps**$]]\!]$ = $[\![$**or**$_2]\!]([\![[_{VP}$ **smoke**$]]\!])([\![[_{VP}$ **sleep**$]]\!])([\![[_{NP}$ **Jeff**$]]\!])$
2. $[\![[_{NP}$ **Jeff**$]]\!]$ = $[\![$**Jeff**$]\!]$ *S2*
3. $[\![[_{VP}$ **smoke**$]]\!]$ = $[\![$**smoke**$]\!]$ *S2*
4. $[\![[_{VP}$ **sleep**$]]\!]$ = $[\![$**sleep**$]\!]$ *S2*
5. $[\![[_{IP}$ **Jeff smoke or sleep**$]]\!]$ = $[\![$**or**$_2]\!]([\![$**smoke**$]\!])([\![$**sleep**$]\!])([\![$**Jeff**$]\!])$ *substitution*
6. $[\![$**or**$_2]\!]$ $([\![$**smoke**$]\!])([\![$**sleep**$]\!])([\![$**Jeff**$]\!])$ = *Lexicon*

$$= \left[\begin{array}{l} \text{f: } (D \rightarrow \{0,1\}) \rightarrow ((D \rightarrow \{0,1\}) \rightarrow (D \rightarrow \{0,1\})) \text{ s.t.} \\ \text{for any } a \in D \text{ and g,h: } D \rightarrow \{0,1\}, \\ f(g)(h)(a) = 1 \text{ iff } g(a) = 1 \text{ or } h(a) = 1 \end{array}\right] (\text{Jeff})(\text{smoke})(\text{sleep})$$

7. $$\left[\begin{array}{l} \text{f: } (D \rightarrow \{0,1\}) \rightarrow ((D \rightarrow \{0,1\}) \rightarrow (D \rightarrow \{0,1\})) \text{ s.t.} \\ \text{for any } a \in D \text{ and g,h: } D \rightarrow \{0,1\}, \\ f(g)(h)(a) = 1 \text{ iff } g(a) = 1 \text{ or } h(a) = 1 \end{array}\right] (\text{Jeff})(\text{smoke})(\text{sleep}) = 1$$

iff Jeff smokes or Jeff sleeps

(23) *Question*: Do rules D1 and D2 allow for the derivation of (24)? If not, why?

(24) Bart or Jeff smoke or sleep

Exercises:

Compute the meaning of the following sentences, and provide the necessary lexical entries:

(25) Sally and Sam sleep

(26) Sam does not sleep

(27) Mary is not tired

4. Two-Place Predicates

Goal: Compositionally derive the truth conditional interpretation of transitive clause (1):

(1) Jeff hit Bart

4.1. Characteristic Functions

So far, intransitive predicates were taken to denote functions from individuals to truth values. But it is also possible to reconstruct predicates as sets of individuals, as is e.g. done in standard predicate logic:

(2) a. $[\![$**smoke**$]\!]$ = $\{x|x \text{ smokes}\}$
b. $[\![$**sleep**$]\!]$ = $\{x|x \text{ sleeps}\}$

On this conception, predicates denote the set of individuals of which the predicate is true. Sets are related to (one place) functions in a systematic way by **CHARACTERISTIC FUNCTIONS**:

(3) For any set A, f is the characteristic function of A iff
for any $x \in A$, $f(x) = 1$ and for any $x \notin A$, $f(x) = 0$

Thus, the two definitions for the one-place predicate **smoke** below are synonymous, because the denotation assigned to **smoke** in (4)a is the characteristic function of the set on the right side of the equation sign in (4)b:

(4) a. $[\![$**smoke**$]\!]$ = $f: D \rightarrow \{0,1\}$ s.t. for any $a \in D_e$, $f(x) = 1$ iff x smokes
b. $[\![$**smoke**$]\!]$ = $\{x|\text{smokes}(x) = 1\}$

For the same reason, the truth values of an expression can be computed either in terms of functions or sets:

(5) a. $[\![$**Bart smokes**$]\!]$ = 1 iff $\left[\begin{array}{l} f: D \rightarrow \{0,1\} \\ \text{for any } a \in D\text{, } f(a) = 1 \text{ iff a smokes} \end{array}\right](\text{Bart}) = 1$

b. $[\![$**Bart smokes**$]\!]$ = 1 iff $[\![$**Bart**$]\!] \in [\![$**smoke**$]\!]$

As it will turn out, it is useful in certain domains to be able to switch from function talk to set talk and v.v. More important for present purposes, characteristic functions serve as a first step in solving the complications posed by transitive predicates for compositionality.

4.2. The Problem

In classical predicate logic, the extension of a two-place predicate could look as follows:

(6) $[\![$**hit**$]\!]$ = $\{\langle\text{Jeff, Bart}\rangle, \langle\text{Jeff, Jeff}\rangle\}$

A pair of individuls satisfies the relation expressed by **hit** if and only if the pair is a member of the denotation in (6). FOr instance, the sentence *Jeff hit Bart* is evaluated as true in this scenario because $\langle\text{Jeff, Bart}\rangle \in \{\langle\text{Jeff, Bart}\rangle, \langle\text{Jeff, Jeff}\rangle\}$. In contrast, the converse sentence *Bart hit Jeff* is false, since <Bart, Jeff> $\notin$ {<Jeff, Bart>, <Jeff, Jeff>}.

In (6), the verb denotation is given as a set, while the present system has used function notation so far. Given the equivalence between functions and sets *via* the concept of the characteristic function, (6) can be rendered by the equivalent formula in (7)

(7) $[\![\textbf{hit}]\!]$ = {<<Jeff, Bart>, 1>, <<Jeff, Jeff>, 1>, <<Bart, Jeff>, 0>, <<Bart, Bart>, 0>,}

It becomes now also possible to define a lexical entry for **hit**, which models **hit** as the characteristic function of the set in (6). More precisely, the 2-place relation below applies to a pair of individuals and returns 1 just in case the pair is a member of the set denoted by **hit.**

(8) $[\![\textbf{hit}]\!]$ = The relation R: D x D $\rightarrow$ {0,1}, such that for any a,b $\in$D,
R(<a,b>) = 1 iff a hit b

There is a problem, though: The principle of compositionality requires that semantic computations proceed in a local, step-by-step fashion by employing Function application. The problem that arises with transitive verbs is that in order to make use of the truth conditional definitions of **hit** above, **hit** would have to apply to the object NP and the subject NP simultaneously. This is so because **hit** is not construed as a function, but as a relation which applies to ordered pairs. This challenge for compositionality can be circumvented by breaking up the relation into the component functions.

4.3. SCHÖNFINKELIZATION

N-place relations can be systematically reduced to 1-place functions by **SCHÖNFINKELIZATION** (Schönfinkel 1924; also referred to as 'Currying' in the literature). There are two ways to *schönfinkel* a two-place relation: left-to-right, and right-to-left:

(9) Jeff hit Bart

(10) *Left-to-Right Schönfinkelization* *('Subject First')*

$$[\![\textbf{hit}]\!] = \begin{bmatrix} \langle\text{Bart, Bart}\rangle & \rightarrow & 0 \\ \langle\text{Bart, Jeff}\rangle & \rightarrow & 0 \\ \langle\text{Jeff, Bart}\rangle & \rightarrow & 1 \\ \langle\text{Jeff, Jeff}\rangle & \rightarrow & 1 \end{bmatrix} = \begin{bmatrix} \text{Bart} & \rightarrow & \begin{bmatrix} \text{Bart} & \rightarrow & 0 \\ \text{Jeff} & \rightarrow & 0 \end{bmatrix} \\ \text{Jeff} & \rightarrow & \begin{bmatrix} \text{Bart} & \rightarrow & 1 \\ \text{Jeff} & \rightarrow & 1 \end{bmatrix} \end{bmatrix}$$

Subject Object

(11) *Right-to-Left Schönfinkelization* *('Object First')*

$$[\![\textbf{hit}]\!] = \begin{bmatrix} \langle\text{Bart, Bart}\rangle & \rightarrow & 0 \\ \langle\text{Bart, Jeff}\rangle & \rightarrow & 0 \\ \langle\text{Jeff, Bart}\rangle & \rightarrow & 1 \\ \langle\text{Jeff, Jeff}\rangle & \rightarrow & 1 \end{bmatrix} = \begin{bmatrix} \text{Bart} & \rightarrow & \begin{bmatrix} \text{Bart} & \rightarrow & 0 \\ \text{Jeff} & \rightarrow & 1 \end{bmatrix} \\ \text{Jeff} & \rightarrow & \begin{bmatrix} \text{Bart} & \rightarrow & 0 \\ \text{Jeff} & \rightarrow & 1 \end{bmatrix} \end{bmatrix}$$

Object Subject

On this conception, transitive verbs are functions from individuals to functions from individuals to truth values. Employing the right-to-left version, it becomes possible to decompose the relation denoted by **hit** into two functions such that the denotation of the verb applies to the denotation of object first, and the VP applies to the denotation of the subject later. This type of analysis observes compositionality, as it computes the meaning of verb and the object prior the meaning of the VP and the NP. Thus, the Fregean denotation of the two-place predicate **hit** is the right-to-left Schönfinkelization of the characteristic function of the relation denoted by **hit**.

In order to compute transitive sentences, one furthermore needs an additional rule for the interpretation of the VP, and a suitable lexical entry for the transitive predicate. The VP rule employs once again function application:

S3. ***V' Rule:*** If α has the form V' (with daughters β γ), then $[\![\alpha]\!] = [\![\beta]\!]([\![\gamma]\!])$

(12) *Lexicon*
$[\![\textbf{hit}]\!]$ = f: D → (D→{0,1}) such that for any a,b ∈D,
f(a)(b) = 1 iff b hit a

<u>Sample Computation</u>

(13)
1. $[\![[_{IP}\ [_{NP}\ \textbf{Jeff}]\ [_{VP}\ \textbf{hit}\ [_{NP}\ \textbf{Bart}]]]]\!]$ = $[\![[_{VP}\ \textbf{hit}\ [_{NP}\ \textbf{Bart}]]]\!]\ ([\![[_{NP}\ \textbf{Jeff}]]\!])$ *S1*
2. $[\![[_{VP}\ \textbf{hit}\ [_{NP}\ \textbf{Bart}]]]\!]$ = $[\![[_{V'}\ \textbf{hit}\ [_{NP}\ \textbf{Bart}]]]\!]$ *S2*
3. $[\![[_{V'}\ \textbf{hit}\ [_{NP}\ \textbf{Bart}]]]\!]$ = $[\![[_{V^\circ}\ \textbf{hit}]]\!]\ ([\![[_{NP}\ \textbf{Bart}]]\!])$ *S3*

4. $[\![[_{IP}\ [_{NP}\ \textbf{Jeff}]\ [_{VP}\ \textbf{hit}\ [_{NP}\ \textbf{Bart}]]]]\!]$ = $[\![[_{V^\circ}\ \textbf{hit}]]\!]\ ([\![[_{NP}\ \textbf{Bart}]]\!])\ ([\![[_{NP}\ \textbf{Jeff}]]\!])$ =
= $[\![\textbf{hit}]\!]$(Bart)(Jeff) = *Substitution, S2 and Lexicon*
5. = $\left[\begin{array}{l}\text{f: D} \rightarrow \text{(D}\rightarrow\text{\{0,1\}) s.t. for any a,b} \in \text{D,}\\ \text{f(a)(b) = 1 iff b hit a}\end{array}\right]$(Bart)(Jeff) = *V°-denotation applies to object*
6. = $\left[\begin{array}{l}\text{g: D}\rightarrow\text{\{0,1\} s.t. for any b} \in \text{D,}\\ \text{g(b) = 1 iff b hit Bart}\end{array}\right]$(Jeff) = *VP-denotation applies to subject*
7. = g: D→{0,1} s.t. g(Bart)(Jeff) = 1 iff Jeff hit Bart

Note that unlike in the relational notation, the Schönfinkelized version of **hit** applies to the object first, and then to the subject. Thus, strict compositionality is observed.

<u>Passive</u>
The example above used the right-to-left Schönfinkelization of **hit**. The alternative left-to-right-Schönfinkelization (10)a yields the passive version of **hit**, in which the *by*-phrase is joined with the verb first, followed by the subject.

(14) $[\![\textbf{hit}_{\textbf{pass}}]\!]$ = (10)a

Exercise

○ Compute the meaning of the example below on the basis of the meaning for **hit** adopted above:

(15) Jeff was hit by Bart

○ What is the Schönfinkelization for *show* given the following scenario (only write down those parts of the table in which the truth function is valued 1)

(16) ⟦**show**⟧ = {<John, Mary, Athens>, <Mary, John, Paris>}

Provide a lexical entry for *show*, which captures its truth-conditions, and design a composition rule. Derive the meaning of *Mary showed John John.*

5. TYPE THEORY

The domain of possible denotations for lexical entries consists now of a variety of functions as well as of individuals. It does not include all types of functions, though. For instance, there are no functions from truth-values to individuals do not have a In order to define this domain more precisely, it is useful to introduce a new tool: type theory.

5.1. RUSSELL'S PARADOX

Russell in a famous letter to Frege obseved in 1902 that Frege's formalization of mathematical logic (more precisely, the system presented in the second volume of *Grundgesetze der Arithmetic*) was inconsistent. Russell's criticism was based sets as defined in (1), which lead to what has become to be known as Russell's Paradox (or Russell-Zermelo Paradox).

(1) $A = \{X \in S | X \notin X\}$
A is the set of all sets which do not themselves as an element.

The Problem

(2) *Question*: Is A a member of A?

- ❒ Assume A is a member of A. Then, A satisfies the restriction of the set, because $X \in S$ and should therefore not be a member of A. The assumption leads to a contradiction.
- ❒ Assume A is not a member of A. Then, A satisfies the condition for membership of the set, because $A \notin A$, and should therefore be a member of A. The assumption leads to a contradiction.

(3) *Answer*: The question cannot be decided, resulting in a paradox

● A standard instantiation of Russell's Paradox looks as follows:

(4) A city council is creating a new position which is meant to improve the appearance of the male population of the city. In particular, the council is sponsoring a barber shop, and employing a barber. Moreover, it is stipulated as a rule that the barber *shaves everyone who does not shave himself*.

(5) *Question*: Does the barber shave himself?

- ❒ If the barber shaves himself, according to the rule he does not shave himself.
- ❒ If the barber does not shave himself, he must abide by the rule and shave himself.

(6) *Answer*: The question cannot be decided, resulting in a paradox.

○ Another version of the paradox which can be found in the literature:

(7) In a library there are many books, some of which are catalogues of books or even catalogues of catalogues. And a catalogue may well list itself, too. Now consider the catalogue of all catalogues that do *not* list themselves: does it list itself or not?

Russell's Solution

Russell proposes to exclude the paradox by introducing a number of well-formed conditions for sets called ***Type Theory*** (Russell 1919). In particular, Russell suggests that sets of the form $\{X \in S | X \notin X\}$ are illicit because 'nothing can contain itself, or be contained in itself'. In a differnt (but logically equivalent) version, this criterion, which Russell refers to as the *vicious-circle principle*, maintains that 'whatever involves all of a collection must not be one of the collection'.[8]

More precisely, type theory establishes a hierarchy of expressions, and states that expressions of the same level can not contain each other. Since predication is expressed in terms of set membership, this in turn entails a restriction on possible predicates, prohibiting predicates such as *to be a set which does not contain itself* or *to shave everyone who does not shave himself.* Russell (incorrectly) thought that this solution would also extend to the Liar Pardox.

There are numerous other semantic paradoxes (or antinomies, as they are sometimes called), among them Grelling's paradox:

Grelling's (1908) Paradox

There are numerous other paradoxes which affect natural language semantics.

(8) a. *autological:* adjectives such that the property the adjective expresses applies to itself (*polysyllabic* [which is polysyllabic], *English* [which is an English word])
 b. *heterological*: adjectives such that the property the adjective expresses does not apply to itself (*monosyllabic* [which is not monosyllabic], *French* [which is not a French word])

Question: Is *heterological* heterological?

❒ Yes: *Heterological* satisfies the definition of autological, and is therefore not heterological, contradicting the assumption.

❒ No: By assumption, *heterological* is therefore autological. Thus, the property which heterologial expresses has to apply to itself. But since this property consists in being heterological, *heterological* is heterological, contradicting the original assumption.

5.2. Functions and Types

Type theory in semantics allows for a less clumsy definition of the domain and range of functions. Following Montague (1973), who introduced type theory in semantics, the domain of types for a language L is made up of e and t, where e stands for individuals, and t for truth values. Derived types are defined in a *recursive* manner, as in A(c) (i.e. the output of A(c) feeds its input):

[8] Generally, the prohibition on Russell-sets is referred to as the *well-foundedness of standard set theory*. There are various other solutions to the paradox, including the axiomatic systems by Zermelo-Fraenkel (ZF) and Neumann-Bernay-Gödel (NBG). More recent developments actually explicitly refute Russell's vicious circle principle (e.g. Peter Azcel's hypersets, as discussed in John Barwise & John Etchemendy. 1992. *The Liar*.)

A. *Semantic Types*

a. e is a semantic type.
b. t is a semantic type.
c. For any semantic types δ, ε, $<\delta, \varepsilon>$ is a semantic type.
d. Nothing else is a semantic type.

B. *Domains*

a. D_e is the domain of individuals.
b. $D_t = \{0,1\}$ is the domain of truth values.
c. For any semantic types δ, ε, $D_{<\delta, \varepsilon>}$ is the domain of functions from D_δ to D_ε.

According to the definitions above, names are assigned type e, the denotations of IP's are of type t, intransitive verbs and adjectives are paired with functions of type <e,t>, and transitive verbs, prepositions and adjectives are interpreted as functions of type <e,<e,t>>. VP, NP, AP and PP-denotations are finally of type <e,t>. Thus, the domain D can be partitioned into various subsets which differ in type: D_t, D_e, $D_{<e,t>,}$ $D_{<e,<e,t>>}$ and so on.

(9) ⟦**love**⟧ = $f \in D_{<e,<e,t>>}$ s.t. for any a, b $\in D_e$, f(a)(b) = 1 iff b loves a

<u>Exercise</u>

What are the types of the following expressions? Provide lexical entries where possible (i.e. not for (10)c, e.g.).

(10) a. ⟦**between**⟧
b. ⟦**brother-of**⟧
c. ⟦**Fish contains phosphor**⟧
d. ⟦**put**⟧
e. ⟦**not**⟧
f. ⟦**yesterday**⟧
g. ⟦**Sally**⟧
h. ⟦**to swim**⟧

5.3. Type Restrictions

Type theory can e.g. be employed to capture restrictions on conjunction. Which are the types of the denotations of the categories which can be conjoined?

(11) a. John is a lawyer and a doctor
b. John is smart and boring
c. John is smart and a lawyer
d. John likes golf and plays tennis
e. John is trying to win the case and anxious to get paid
f. John is trying to win the case and an example of a lawyer

(12) a. *Mary [[$_{V°}$ ate] and [$_{V°}$ slept]] pizza
b. *Mary saw [[$_{CP}$ John playing the guitar] and [$_{NP}$ a lawyer]]
c. *Mary played [[$_{NP}$ the guitar] and [$_{PP}$ in the afternoon]]

Consider the interpretation of **lawyer** and **smart**: both expressions are one-place predicates, and are therefore of type <e,t>. **play** and **eat** are of type <e,<e,t>>, and clauses denote truth-values.

(13) *Conjecture:* Only expressions of the same type can be conjoined (Partee & Rooth 1983).

---

6. Type Driven Interpretation

6.1. Presuppositions

Presuppositions[9] are entailments which are triggered by lexical items, and which determine the contexts in which an utterance can be felicitously used.

(1) a. Speaker A: "Mary stopped drinking"
b. Presupposition: Mary used to drink

For instance, using (1)a even though the speaker knows that Mary has never drunk results in a presupposition failure and an infelicitous discourse. (Such presupposition failures can be repaired, though, by reconstructing counterfactual contexts in which Mary drank, and by using Gricean maxims in order to account for the fact that the speaker conveys counterfactual instead of actual information).

Exercise

What happens if the presuppositions of (parts of) a clause are not met? Are such sentences false? Assume that Mary has never drunk. In such a scenario, are the following statements true or false? (Three valued logics and supervaluation logics discusses such issues (Keefe and Smith 1997).)

---

[9]One way to define presuppositions can be found in Chierchia and McConnell-Ginet (1990).
(i) The *common ground* is the set of propositions speaker and hearer share a belief in.
(ii) A sentence S *presupposes* a proposition p iff in any context c where S has a semantic value relative to c, p follows from the common ground of c.

(2) a. Mary stopped drinking
b. It is not the case that Mary stopped drinking
c. Mary stopped drinking and it is not the case that Mary stopped drinking

6.2. Definite Descriptions ('Kennzeichnungen')

Traditionally, one refers by the name *definite description* to phrases of the shape 'the F'. But there are a number of NPs which do not satisfy this surface criterion, but still function as definite descriptions semantically.

● NPs modified by prenominal possessives are definite descriptions:

(3) a. She bought Sally's house
b. She bought the house of Sally's

(4) a. She bought his house
b. She bought the house of his

● For Russell, who worked out the most famous theory of descriptions (see below), names are also hidden definite descriptions. The nam *Betrand Russell* can e.g. be assigned the logical form representation *the English logician and philosopher who was born in 1872 and died in 1970 who wrote Principia Mathematica, ...* This claim is generally thought to be problematic, though.

● Some types of pronouns - sometimes called pronouns of laziness - can be analyzed as hidden definite descriptions:

(5) a. A man entered the room. He sat down
b. A man entered the room. [The man who entered the room] sat down

○ Note that not every pronoun can be interpreted as a lazy pronoun. (6)a and (6)b do not mean the same, and (7) is ungrammatical:

(6) a. Every boy owns a bicycle. He rides it to school.
b. Every boy owns a bicycle. The boy who owns a bicycle rides it to school.

(7) *No boy$_k$ owns a bicycle. He$_k$ rides it to school.

6.2.1. The Three Ingredients of Definite descriptions

(8) The **P**resident **o**f **t**he **U**nited **S**tates ('POTUS') is bald.

The different contributions of the definite description to the overall meaning in sentences such as (8) can be broken down into three components. When uttering (8), one makes three different assertions:

(9) (i) There is a POTUS.
(ii) At most one thing is a POTUS.
(iii) Everything which is a POTUS is bald.

(i) is the *existence assertion,* (ii) *the uniqueness assertion,* and (iii) represents the predication relation. Out of these three components, the existence and uniqueness criteria are construction specific to the interpretation of definite descriptions.

6.2.2. A Fregean presupposition account

The classic example (10) is judged as somehow infelicitous since there is no King of France today, and the statement therefore violates the existence condition of definite description. The infelicity of (11)b is on the other side due to the fact that a member by definition has two members, which conflicts with the uniqueness requirement imposed by *the.*

(10) #The king of France is bold

(11) a. Sam remembered only a member of the pair
b. #Sam remembered only the member of the pair

What is now the semantic contribution of *the*? Note to begin with that the definite determiner combines with one-place predicates:

(12) a. They met the Pope
b. *They met the Johannes

The existence and the uniqueness condition can be expressed as presuppositions. That is, instead of writing these existence and uniqueness into the truth conditions for **the** (on this option see below), these conditions are interpreted as restrictions on the domain of **the**:

(13) $[\![\textbf{the}]\!]$ = $f \in D_{<<e,t>e,>}$: for any $g \in D_{<e,t>}$ s.t. there is exactly one $a \in D_e$ s.t $g(a) = 1$, $f(g)$ = the unique a for which $g(a) = 1$

But with this modification, **the** is no longer a function of type <<e,t>,e>, i.e. a function from the whole domain of predicates (<e,t>). Rather, the denotation of **the** is a so-called PARTIAL FUNCTION. Partial functions are functions which are not defined for the whole domain, but only for a subset thereof. In the case at hand, the domain of **the** is not the whole set $D_{<e,t>}$ but only the subset $A \subseteq D_{<e,t>}$ such that for each member f of A, there is a single individual a such that $f(a) = 1$. Thus, **the** is defined only for this subset A.

6.2.3. Russell's theory of descriptions

The general format for definite descriptions looks as in (14):

(14) a. The F is a G
b. (i) There is an F
(ii) At most one thing is an F
(iii) Everything which is F is G

Russell proposed the following non-presuppositional interpretation (Russell, 1905, *On Denoting*. Mind 14. 479-493):

(15) *Russellian definite description*$_{Def}$

$$[\![\text{The F is G}]\!] = \underbrace{\exists x[F(x)}_{\text{'there is an F' (i.e. x)}} \wedge \underbrace{\forall y[F(y) \rightarrow x = y]}_{\text{'at most one thing [= x] is an F'}} \wedge \underbrace{G(x)]}_{\text{'everything which is F [= x] is G'}}$$

A shorter, logically equivalent version reads as follows:

(16) $\exists x[\forall y[F(y) \leftrightarrow x = y] \wedge G(x)]$

○ Applying definition (15) to example (8) yields e.g. the semantic representation below:

(17) a. Let F be instantiated by *POTUS* and
let G be instantiated by *bald*

b. $[\![\text{The POTUS is bald}]\!] = \exists x[\text{POTU}(x) \wedge \forall y[\text{POTUS}(y) \rightarrow x = y] \wedge \text{bald}(x)] =$

c. $= \exists x[\forall y[\text{POTUS}(y) \leftrightarrow x = y] \wedge \text{bald}(x)]$

● Russells theory has a number of consequences, two of which are listed below.

<u>I. Non-referring expressions</u>

(18) #The King of France is bald

Russell's Solution: In all situations without a King of France, (18) is a false statement

(19) $\exists x[\text{King of France}(x) \wedge \forall y[\text{King of France}(y) \rightarrow x = y] \wedge \text{bald}(y)]$

<u>II. Negative existence assertions</u>

(20) The Tower of Babel did not exit

The problem posed by (20) presents itself in the following form: Assume that the Tower of Babel existed. Then (20) states that it did not exist, resulting in a contradiction. Assume alternatively that it did not exist. Then (20) predicates (= expresses) something about an object which does not exist. But this seems to be impossible.

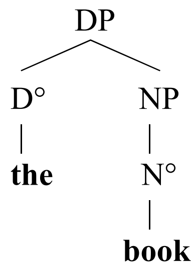
Russell's Solution: Negation can have narrow scope ((21)a) or wide scope ((21)b). In the former reading, sentence leads to contradiction. In the latter reading, sentence expresses a contingent statement.

(21) a. $\exists x[\text{Tower of Babel }(x) \wedge \forall y[\text{Tower of Babel}(y) \rightarrow x = y] \wedge \neg\text{exist}(y)]$
b. $\neg\exists x[\text{Tower of Babel }(x) \wedge \forall y[\text{Tower of Babel}(y) \rightarrow x = y] \wedge \text{exist}(y)]$

6.3. ABANDONING CONSTRUCTION SPECIFIC RULES

Even though (13) provides a suitable lexical entry for **the**, it is not possible to compute sentences containing definite descriptions, yet. This is so because the fragment contains no composition rule which would allow to interpret trees such as:

(22)



Of course, it would be possible to write a separate rule for DP's. However, following this strategy would require to design a designated composition rule for all non-terminal branching nodes (DP, PP, IP, CP, AP, DegP, ...), resulting in a rather awkward system. Type theory offers a more elegant solution, which rests on the assumption that any two sister nodes which meet the type requirements are interpreted by Function application

FA. Function application:	If α has two immediate daughter nodes β and γ (in any order), and $[\![\beta]\!] \in D_{<\delta, \varepsilon>}$ and $[\![\gamma]\!] \in D_{\delta}$, then $[\![\alpha]\!] = [\![\beta]\!]([\![\gamma]\!])$

NB: Apart from FA, one still needs the rule for non-branching nodes S2.) In general, letting interpretation be driven by the principles of type theory generates a parsimonious syntax-semantics interface which - in most cases - does need to include explicit interpretation rules which refer to category labels. On this conception, semantics 'sees' only denotations and structure, but is agnostics as to the actual category labels.

Question: Do the following examples pose a problem for the hypothesis that all selectional restrictions are captured by semantic principles?

(23) a. *Mary is the old
b. *Jeff man
c. *Blue Sam is

6.4. THE Θ-CRITERION AND INTERPRETABILITY

(24) *Θ-Criterion*
a. Every referential NP is assigned exactly one Θ-role.
b. Every Θ-role is assigned to exactly one referential NP.

● Examples which are blocked by (24)a and (24)b: Sentence (24)b is excluded either by the assumption that the NP *Sam* is assigned less than one Θ-role, or by the assumption that the subject Θ-role is assigned to more than one referential NP.

(25) *Ann arrived Sam.

● Examples which are blocked by (24)a, but not by (24)b:

(26) Ann loved
  a. possible reading: $\exists x[\text{love}(x)(\text{Ann}) \wedge \neg \text{Ann} = x]$
  b. impossible reading: love(Ann)(Ann) *Subject is assigned more than one Θ-role*

● Examples which are blocked by (24)b, but not by (24)a:

(27) *arrived *Subject Θ-role is assigned to less than one referential NP*

The Θ-Criterion can now also be given a semantic interpretation: VP's denote functions of type <e,t> or <e,<e,t>>, and sentences denote truth values. Moreover, assume the principle of Full Interpretation (Chomsky 1986, 1995), according to which all expressions in a clause have to be assigned an interpretation. Given these premises, clauses embedding spurious arguments contain terms whose denotations cannot be integrated into the computation:

(28) *Ann arrived Sam

(29) a. $[\![\textbf{arrived Sam}]\!] = [\![\textbf{arrived}]\!]_{<e,t>}([\![\textbf{Sam}]\!]_e) =$ 1 iff Sam arrived
  b. $[\![\textbf{Ann arrived Sam}]\!] = [\![\textbf{arrived Sam}]\!]_t\ [\![\textbf{Ann}]\!]_e$ ☠ *Type mismatch*

Moreover, if a predicate lacks one of its arguments, the denotation of the sentence will no longer be a truth value, but an open formula:

(30) a. *arrived
  b. $[\![\textbf{arrive}]\!] = f \in D_{<e,t>}$

Finally, examples in which one NP is assigned more than one Θ-role are excluded by the assumption that there is a one-to-one correspondence between the number of arguments of a relation and its arity (number of arguments a relation takes):

(31) Ann loved
  a. possible reading: $\exists x[\text{love}(x)(\text{Anne})]$
  b. impossible reading: love(Ann)(Ann) *Subject is assigned more than one Θ-role*

Thus, the Θ-Criterion can be reduced to the principles of semantic interpretability.

7. MODIFICATION

● AP's and NP's that instantiate non-verbal predicates denote functions from D to {0,1}:

(1) a. $[\![$**blue**$]\!]$ = f: D → {0,1} s.t. for any a∈D, a is blue
b. $[\![$**book**$]\!]$ = f: D → {0,1} s.t. for any a∈D, a is a book

○ Adjectives and prepositions that denote non-verbal predicates from D to D→{0,1}:

(2) a. Sam is fond of Mary
b. Sally is in Uppsala

(3) a. $[\![$**fond**$]\!]$ = f: D → (D → {0,1}) s.t. for any a, b∈D
f(a)(b) = 1 iff b is fond of a

b. $[\![$**in**$]\!]$ = f: D → (D → {0,1}) s.t. for any a, b∈D
f(a)(b) = 1 iff b is in a

Question: What is the translation of the copula **be** in (2)? Can it always be treated this way?

<u>Exercise</u>
Devise lexical entries for **above** (as in 'Vienna is above the sea level'), **between** (as in "3 is between 2 and 4"), **during** (as in 'She arrived during the break') and **taller** (as in 'John is taller than Mary').

7.1. PREDICATE MODIFICATION

Goal: Compositionally derive the interpretation of attributive modifiers.

(4) The blue book

(5) a. $[\![$[$_{IP}$ [$_{NP}$ **Sam**] [$_{VP}$ **read** [$_{DP}$ **the** [$_{NP}$ [$_{AP}$ **blue**][$_{NP}$ **book**]]]]]$]\!]$ = ?
[$_{NP}$ [$_{AP}$ **blue**][$_{NP}$ **book**]] = {x|x is blue and x is a book}

PM. Predicate Modification

○ If α has the form $\underset{\beta \quad \gamma}{\mathrm{XP}}$, and $[\![\beta]\!], [\![\gamma]\!] \in D_{<e,t>}$ then $[\![\alpha]\!] = [\![\beta]\!] \cap [\![\gamma]\!]$ *Set notation*

○ If α has the form $\underset{\beta \quad \gamma}{\mathrm{XP}}$, and $[\![\beta]\!], [\![\gamma]\!] \in D_{<e,t>}$ *Function notation*

then $[\![\alpha]\!]$ = the function $f_{<e,t>}$ such that for any $a \in D_e$,
f(a) = 1 iff $[\![\beta]\!]$(a) = 1 and $[\![\gamma]\!]$(a) = 1

Sample Computation

(6) 1. $[\![[_{IP}\ [_{NP}\ \textbf{Sam}]\ [_{VP}\ \textbf{read}\ [_{DP}\ \textbf{the}\ [_{NP}\ [_{AP}\ \textbf{blue}\ ][_{NP}\ \textbf{book}]]]]]]\!]$ = *FA*

$= [\![[_{VP}\ \textbf{read}\ [_{DP}\ \textbf{the}\ [_{NP}\ [_{AP}\ \textbf{blue}\ ][_{NP}\ \textbf{book}]]]]]\!]([\![[_{NP}\ \textbf{Sam}\ ]]\!])$ =

2. $= [\![[_{V^\circ}\ \textbf{read}]]\!]\ ([\![[_{DP}\ \textbf{the}\ [_{NP}\ [_{AP}\ \textbf{blue}\ ][_{NP}\ \textbf{book}]]]]\!])([\![[_{NP}\ \textbf{Sam}\ ]]\!])$ = *FA*

3. $= [\![\textbf{read}]\!]\ ([\![[_{DP}\ \textbf{the}\ [_{NP}\ [_{AP}\ \textbf{blue}\ ][_{NP}\ \textbf{book}]]]]\!])([\![\textbf{Sam}\ ]\!])$ *S2*

4. $[\![[_{DP}\ \textbf{the}\ [_{NP}\ [_{AP}\ \textbf{blue}\ ][_{NP}\ \textbf{book}]]]]\!] = [\![\textbf{the}]\!]\ ([\![[_{NP}\ [_{AP}\ \textbf{blue}\ ][_{NP}\ \textbf{book}]]]]\!])$ *FA & S2*

5. $[\![[_{NP}\ [_{AP}\ \textbf{blue}\ ][_{NP}\ \textbf{book}]]]]\!]$ = *PM & Lexicon*

$= f \in D_{<e,t>}$ s.t. for any $a \in D_e$ blue(a) = 1 and book(a) = 1

6. $[\![\textbf{the}]\!]\ ([\![[_{NP}\ [_{AP}\ \textbf{blue}\ ][_{NP}\ \textbf{book}]]]]\!])$ = *Lexicon* ('the')

$$= \begin{bmatrix} f \in D_{<<e,t>,e>} \text{ for any } \boxed{g \in D_{<e,t>}} \text{ s.t. there is exactly one } a \in D_e \\ \text{s.t } g(a) = 1,\ f(g) = \boxed{\text{the unique a for which } g(a) = 1} \end{bmatrix} \boxed{\begin{array}{l}\text{(the function } f_{<e,t>} \\ \text{s.t. for any } a \in D_e, \\ f(a) = 1 \text{ iff blue(a)} = \\ 1 \text{ and book(a)} = 1)\end{array}}$$

NB: The function, which is defined for any $g \in D_{<e,t>}$ applies to the boxed argument on the right-hand side. The value of the function is specified in the box in the second line.

7. = if defined: the unique a for which blue(a) = 1 and book(a) = 1 *FA*

NB: The 'if define' part is a short-hand for stating that the function only returns a value if there is exactly one $a \in D_e$ s.t. g(a) = 1. This condition is the presupposition of the definite description.

8. $[\![\textbf{read}]\!]\ ([\![[_{DP}\ \textbf{the}\ [_{NP}\ [_{AP}\ \textbf{blue}\ ][_{NP}\ \textbf{book}]]]]\!])([\![\textbf{Sam}\ ]\!])$

$$= \begin{bmatrix} f_{<e,<e,t>>} \text{ s.t. for any a,b } \in D_e, \\ f(a)(b) = 1 \text{ iff b read a} \end{bmatrix} \text{(the unique a for which blue(a) = 1 and book(a)) (Sam)}$$

9. $$\begin{bmatrix} f_{<e,<e,t>>} \text{ s.t. for any a,b } \in D_e, \\ f(a)(b) = 1 \text{ iff b read a} \end{bmatrix} \text{(the unique a for which blue(a) = 1} \wedge \text{book(a))(Sam)} = 1$$

iff Sam read the unique a for which blue(a) = 1 and book(a) = 1

7.2. Types of Modification

Traditionally, linguistic research distinguishes among two types of modification:

A. Restrictive modifiers: co-determine the denotation of the expression they modify and thereb help to identify the extension of an NP. They can be instantiated by attributive APs and restrictive relative clauses. A subgroup of these constructions, the so-called INTERSECTIVE modifiers (see below) can be treated in terms of set intersection:

(7) $[\![\textbf{blue book}]\!]$ = {x|x is blue and x is a book}
(8) $[\![\textbf{book which Bill read}]\!]$ = {x|x is a book and Bill read x}

Other intersective adjectives include *Swedish, square, pink, sick,*

B. Non-restrictive modifiers: do not contribute to fixing the denotation of the expression they modify, but merely add additional qualifications:

(9) Bill, who (- by the way -) passed the test yesterday, will be in Paris tomorrow.

(10) The number π, whose importance has been recognized by the Babylonians and Egyptians, is irrational.

(11) Sally will leave on Tuesday, which coincidentally is also Bart's birthday.

Restrictive and non-restrictive interpretation sometimes correlate with word order variation (on this type of phenomenon in Romance see e.g. Bernstein 1993: 24f):

(12) a. las olorosas flores *non-restrictive*
"the flowers, who (by the way) smell"
b. las flores olorosas *restrictive*
"the flowers who smell"

7.2.1. Three types of Adjectives

From a logical point of view, AP denotations can be grouped into three classes.

I. Intersective adjectives

Intersective adjectives form APs which satisfy the intersection criterion in (13), (21):

(13) *The Property of Intersectivity*

⟦AP NP⟧ = ⟦AP⟧ ∩ ⟦NP⟧

e.g. ⟦square table⟧ = ⟦square⟧ ∩ ⟦table⟧

II. Non-intersective and Subsective adjectives

Adjectives such as a **leftmost** cannot be treated as intersective modifiers. In the case of **leftmost**, intersective interpretation yields the wrong truth-conditions.

(14) a. the leftmost black circle
b. the circle which is leftmost and which is black

(15) a. ①❷❸❹❺⑥ {x|leftmost(x)}∩{x|black(x)}∩{x|circle(x)} = ∅
b. ❶②③❹❺⑥ {x|leftmost(x)}∩{x|black(x)}∩{x|circle(x)} = ❶

Unlike intersective adjectives (see (16)), NPs attributively modified by non-intersective adjectives such as **alleged, perfect**, **skillful**, **fake** and **former** cannot be decomposed into two conjuncts *salva veritate,* i.e. in a meaning preserving way. To say that a group of people are *perfect cooks* ((16)ba) is not the same as asserting that they are perfect and that they are cooks: they are perfect only in on respect, i.e. as cooks.

(16) a. They are Danish cooks =
b. = They are Danish and they are cooks

(17) a. They are perfect cooks
b. ≠ They are perfect and they are cooks

(18) ⟦perfect cooks⟧ ≠ ⟦perfect⟧ ∩ ⟦cooks⟧ *non-intersective*

● Moreover, if adjectives such as **alleged** were intersective, the following entailment relation should hold:

(19) If ⟦α⟧ = ⟦β⟧ then ⟦**alleged** α⟧ = ⟦**alleged** β⟧

But clearly, the inference in (20) below is invalid:

(20) The murderer was the president of the company
She met the alleged murderer
∴ She met the alleged president of the company

○ Formally, non-intersective adjectives fall into two groups. First, there are adjectives which are subsective, as defined in (21), among them **perfect**, **skillful**, **good**, and **recent**:

(21) *The Property of Subsectivity*
⟦AP NP⟧ = ⟦AP⟧ ⊆ ⟦NP⟧
e.g. ⟦perfect cook⟧ = ⟦perfect⟧ ⊆ ⟦cook⟧

If an individual is a perfect cook, then he is perfect as a cook, but not necessarily as a composer, writer, etc... Thus, in this example, 'perfect' is only a property of cooks, and not of other groups of individuals.

III. Non-intersective and Non-Subsective adjectives

Finally, there are adjectives which are non-intersective, but also fail to observe the subsective property. This last group includes **former, alleged, fictitious, putative, imaginary,arguable,** and **counterfeit**.

(22) They are alleged/former criminals
a. ⟦alleged/former criminals⟧ ≠ ⟦alleged/former⟧ ∩ ⟦criminals⟧ *non-intersective*
b. ⟦alleged/former criminals⟧ ⊄ ⟦criminals⟧ *non-subsective*

Its members can only show up in attributive position:

(23) a. They are alleged criminals
b. *They are alleged and they are criminals

This observation serves as a further argument against analyzing the nexus between **alleged** and the common noun they modify by intersection. If **alleged** were intersective, it would remain mysterious why it cannot be construed in predicative position.

Note finally that if an adjective is intersective, it is also subsective, but not v.v.

Exercise

Show why this is the case.

7.2.2. Two Open Issues

I. Relative Clauses: may serve as restrictive modifiers, which semantically are intersected with the common noun denotation:

(24) a. $[\![[_{NP}\ \textbf{the book}\ [_{CP}\ \textbf{which t is blue}]]]\!]$ = $\{x|\text{book } x \text{ and } x \text{ is blue}\}$
b. $[\![[_{NP}\ \textbf{the book}\ [_{CP}\ \textbf{which Bill read t}]]]\!]$ = $\{x|\text{book } x \text{ and Bill read } x\}$

The relative clause internal trace is translated here as a semantic variable (i.e. *x* in the restriction of the set). Assume moreover that the relative pronoun is semantically vacuous (as can be seen from the fact that it is optional in some contexts).

(25)
1. $[\![\textbf{t}]\!]$ = x
2. $[\![\textbf{read x}]\!]$ = $[\![\textbf{read}]\!](x)$
3. $[\![\textbf{which Bill read x}]\!]$ = $[\![\textbf{read}]\!](x)(\text{Bill})$
4. $[\![\textbf{read}]\!](x)(\text{Bill})$ = $\begin{bmatrix} f_{<e,<e,t>>} \text{ s.t. for any } a,b \in D_e, \\ f(a)(b) = 1 \text{ iff } b \text{ read } a \end{bmatrix}(x)(\text{Bill})$
5. $[\![\textbf{book which Bill read x}]\!]$ = $[\![\textbf{book}]\!]_{<e,t>}\ [\![\textbf{which Bill read x}]\!]_{<t>}$ ☠ *Type conflict*

The computation crashes in line 5, because the common noun denotes a predicate, but the relative clause denotes a **formula** (an expression of type t). Thus, the relative clause and the NP cannot be composed by the principles licit under the assumption of type driven interpretation.

Goal I: Find a tool to represent relative clauses as predicates.

II. Pronouns: In contrast to names such as *Ann,* the reference of pronouns can vary from situation to situation. With the rules introduced so far, pronouns cannot be interpreted.

(26) a. Ann slept
b. She slept

Goal II: Define a suitable semantic characterization of pronominal reference.

It will turn out that the strategies which will lead to a better understanding of relative clauses and pronouns are related.

8. The λ-Calculus

In mathematics, functions express correpondences between two variables, the dependent and the indepent one. The expression in (27) denotes the value of a function. If f is e.g. (extensionally) defined as in (28), the value is '$x^2 + 1$', which denotes an object of type e (assuming that numbers are of type e). The function can therefore also be specified by the formula in (29):

(27) f(x)

(28) f: $1 \rightarrow 2$
$2 \rightarrow 5$
$3 \rightarrow 10$
$4 \rightarrow 17$
....

(29) $f(x) = x^2 + 1$

The classical function notation is not optimally transparent, though. It does not permit to make reference to the function itself. There is no notational device that would allow to refer to the abstract object (of type <e,e>) representing the operation that maps 1 to 2, 2 to 5, 3 to 10 and so on. This is so because in the classical mathematical notation, the function symbol ('f') cannot be separated from the independent variable. Expressions such as '$f = x^2 + 1$' are not well-formed.

The λ-calculus, conceived by Alonso Church (1936), fills this gap (see Church 1940, Curry 1934 for typed λ-calculus). In particular, the λ-calculus makes it possible to seperate the function from the variable, and thereby provides a way to refer to the name f independently of the variable x. The symbol 'f' on the left-hand side of (30) unambiguously specifies a function, in this case an object of type <e,e>. The right-hand side of (30) is called a λ-term (on 'term' see also fn. 4).

(30) $f = \lambda x[f(x)]$

Thus, (27) (by convention) denotes the value of a function, while the function itself is represented by the λ-term (30).

<u>Example</u>

The expressions under a. can be expressed in λ-calculus as in b:

(31) a. $f(x) = x^2 + 1$
b. $f = \lambda x[x^2 + 1]$

(32) a. $f(x,y) = x + y$
b. $f = \lambda x \lambda y[x + y]$

Montague (1970) was the first to apply the (typed version of the) λ-calculus to the analysis of NL.

○ For an updated version of this part see http://users.uoa.gr/~wlechner/Sem2013.htm

○ An interactive guide with online exercises and solutions has been provided by Chris Barker at http://ling.ucsd.edu/~barker/Lambda/

○ For a more demanding introduction by Barendregt and Barendsen see: http://www.cs.ru.nl/home/Erik.Barendsen/onderwijs/T3/materiaal/lambda.pdf

8.1. λ-ABSTRACTION

Illustrating λ-abstraction by means of the example in (1), the λ-operator turns an OPEN FORMULA (i.e. an expression of type t which contains an unbound variable) into a derived predicate (type <e,t>) by abstraction over the free variable.

(1) read(x)(Bill) *open formula*

(2) a. which_x Bill read t_x
b. λx[read(x)(Bill)] *λ-abstraction over x* ⇨ *predicate*

The operation of λ-abstraction is defined in a more general way such that it can be applied to expressions of any type. That is, the variable followed by the λ-operator can range over any possible type (predicates, relations, functions,...). As any formal language, the λ-calculus is precisely defined by a syntax, which generates all and only the well-formed expressions of the calculus, and a semantics interpreting these expressions.

8.1.1. Syntax for λ-terms

(3) If φ is an expression of type δ, and α is a variable of type ε, then λα [φ] is an expression of type <ε,δ>.

(4) a. f(x) = x+2
b. y = x+2
c. λx[x+2] *the function which maps arbitrary x to x+2*

(5) a. love(x)(y)
b. λx[love(x)(John)] *the individuals John loves*
c. λxλy[love(x)(y)] *two-place relation of 'love'*

8.1.2. A remark on notation

In the literature, there are a number of different ways to write λ-terms. For reasons of visibility (on the blackboard) we will use square brackets to mark the scope of the λ-term. Other authors employ round brackets, and in the classical λ-calculus, the λ-operator and the body or dots (.).

(6) λxλy[see(x)(y)] ≈ λxλy(see(x)(y)) ≈ λxλy.see(x)(y)

8.1.3. Semantics for λ-terms

There are two possible ways of interpreting λ-terms

(7) a. λα [φ]: "the smallest function which maps α to φ."
b. λα [φ]: "the function which maps α to 1 if φ, and to 0 otherwise"

Choice between these versions depends on the context. In general, reading (7)a applies if φ does not denote a truth value, and (7)b if φ denotes a truth value.[10] When φ values over $\{0,1\}$, the λ-expression $\lambda\alpha[\varphi]$ is a characteristic function for a set, and can therefore also serve as notation for a set.

(8)

	Set Notation	*λ-Notation*
a.	$\{x \in D \mid smoke(x) = 1\}$	$\lambda x[smoke(x)]$ "the function which maps x to 1 if x smokes, and to 0 otherwise" ≈ set of smokers
b.	$\{x \in D \mid love(x)(John) = 1\}$	$\lambda x[love(x)(John)]$ "the function which maps x to 1 if John loves x, and to 0 otherwise" ≈ set of individuals John loves
c.	$\{x \in D \mid \{y \in D \mid love(x)(y) = 1\}\}$	$\lambda x \lambda y[love(x)(y)]$ "the function which maps x to the function which maps y to 1 if y loves x, and to 0 otherwise"

(9)

		Examples	*Type*
a.	$\lambda x[P(x)]$	$\{x \in D \mid P(x) = 1\}$	$\langle e,t \rangle$
b.	$\lambda P[P(a)]$	$\{P \in D_{\langle e,t \rangle} \mid P(a) = 1\}$	$\langle\langle e,t \rangle,t \rangle$
c.	$\lambda x \lambda y[R(x)(y)]$	$\{x \in D \mid \{y \in D \mid R(x)(y) = 1\}\}$	$\langle e,\langle e,t \rangle\rangle$
d.	$\lambda R \lambda Q[R(y)(x) \vee Q(y)]$	....	$\langle\langle e,\langle e,t \rangle\rangle,\langle\langle e,t \rangle,t \rangle\rangle$
e.	$\lambda \wp \lambda x \lambda R \lambda y[\wp(R(x)) \wedge R(y)(x)]$		$\langle\langle\langle e,t \rangle,t \rangle,\langle e,\langle\langle e,\langle e,t \rangle\rangle,\langle e,t \rangle\rangle\rangle\rangle$

Note that the order of abstraction matters! Consider the two order in which abstraction over x and y can be applied to love(x)(y):

(10) *Abstracting over x first*

a. $\lambda x[love(x)(y)]$ — *the individuals y loves/ 'being loved by y' relation*
b. $\lambda y \lambda x[love(x)(y)]$ — *two-place relation of 'being loved by'*

(11) *Abstracting over y first*

a. $\lambda y[love(x)(y)]$ — *the individuals which love x/ 'loving x' relation*
b. $\lambda x \lambda y[love(x)(y)]$ — *two-place relation of 'love'*

Exercise

● What is the type of the following expressions (where x, y,a,d $\in D_e$, and P, M$\in D_{\langle e,t \rangle}$). What is the type of Q and R? (Assumption: all expressions inside square brackets are of type t)

(12)
a. $\lambda P[P(x) \wedge M(x)]$
b. $\lambda P \lambda M \exists x[P(x) \vee \neg M(x)]$
c. $\lambda x \lambda R \lambda P[R(P)(x)]$
d. $\lambda M \lambda x \lambda P[\neg P(x) \wedge M(x)]$
e. $\lambda x \lambda y \lambda P \lambda d[Q(P)(d)(x) \wedge M(y)]$
f. $\lambda Q \lambda R \lambda x \lambda M[\neg Q(R(M))(x)(a)]$

[10] Both readings can be subsumed under (7)a, given that λx[**smoke**(x)] is read as "the function which maps x to the denotation of **smoke**(x)" (NB: **smoke**(x) denotes a truth value).

○ Which of the following expressions are well-formed? If a term is well-formed, provide its type (assume a and b are individual constants, that $P,M \in D_{<e,t>}$, $Q \in D_{<e,<e,t>>}$ and that $\wp \in D_{<<e,t>,t>}$):

(13)
a. $\lambda x[P(a)]$
b. $\lambda x[M(x)] \vee M(b)$
c. $\lambda x[M(x) \vee M(b)]$
d. $\lambda \neg P \lambda y[P(y)]$
e. $\lambda P \lambda x[P(a)(b)]$
f. $\lambda Q \lambda x[Q(a)(x) \wedge P(b)]$
g. $\lambda M \lambda \wp[\wp(M) \rightarrow \neg M(b)]$
h. $\neg \lambda Q[Q(b)(a)]$

● What is the meaning of expressions such as $\lambda x[P(y)]$ or $\lambda x[Q(2)]$?

8.2. λ-CONVERSION

λ-terms can be reduced by three rules, which as a group are referred to as λ-conversion. The two which are relevant for present purposes are α-conversion and β-reduction

α-conversion: formalizes the fact that λ-expressions which only differ in the name assigned to bound variables (alphabetic variants) are freely interchangeable. The λ-expressions below all denote the same function:

(14)
$\lambda x \lambda y[F(x)(y)] \quad \equiv$
$\equiv \lambda w \lambda z[F(w)(z)] \quad \equiv$
$\equiv \lambda y \lambda x[F(y)(x)] \quad ...$ *α-conversion*

β-conversion (or β-reduction): expresses function application, i.e. the reduction of an expression by applying the function to one of its arguments. Generally, the term *λ-conversion* is used in a narrow interpretation as referring to β-reduction.

(15) $\lambda \mathbf{x}[\text{read}(\mathbf{x})(\text{Bill})](\mathbf{Kafka}) \quad = \quad \text{read}(\mathbf{Kafka})(\text{Bill})$ *β-reduction/λ-conversion*

(16) If $\lambda\alpha\ [\varphi]$ is an expression of type $<\varepsilon,\delta>$, and γ is an expression of type ε, then $\lambda\alpha\ [\varphi](\gamma)$ is the expression of type δ which results from substituting all occurrences of α in φ by γ. (Slightly more formally: $\lambda\alpha\ [\varphi](\gamma) = \varphi^{[\alpha \rightarrow \gamma]}$)

	Examples			*Type*
(17)	a. $\lambda x[P(x)](a)$	=	$P(a)$	t
	b. $\lambda P[P(a)](Q)$	=	$Q(a)$	t
	c. $\lambda x \lambda y[P(x)(y)](a)$	=	$\lambda y[P(a)(y)]$	<e,t>
	d. $\lambda x \lambda y[P(x)(y)](a)(b)$	=	$\lambda y[P(a)(y)](b)$	<e,t>
		=	$P(a)(b)$	t

Note that the understood bracketing of $\lambda x \lambda y[P(x)(y)](a)(b)$ is $\lambda x[\lambda y[P(x)(y)]](a)(b)$ and not $\lambda x[\lambda y[P(x)(y)](a)](b)$, unless otherwise indicated. This is a common source of confusion!

(18)

	Set Notation		*λ-Notation*
a. $[\![\text{love}]\!] \rightarrow$			
	$\{x \mid \{y \mid \text{love}(x)(y)\}\}$		$\lambda x \lambda y[\text{love}(x)(y)]$
b. $[\![\text{love Peter}]\!] \rightarrow$			
	$\mathbf{Peter} \in \{\mathbf{x} \mid \{y \mid \text{love}(\mathbf{x})(y)\}\} =$		$\lambda \mathbf{x} \lambda y[\text{love}(\mathbf{x})(y)](\mathbf{Peter}) =$
$=$	$\{y \mid \text{love}(\mathbf{Peter})(y)\}$	$=$	$\lambda y[\text{love}(\mathbf{Peter})(y)]$ *(by conversion)*
c. $[\![\text{They love Peter}]\!] \rightarrow$			
	$\mathbf{they} \in \{\mathbf{y} \mid \text{love}(\text{Peter})(\mathbf{y})\}$		$\lambda \mathbf{y}[\text{love}(\text{Peter})(\mathbf{y})](\mathbf{they}) =$
$=$	$\text{love}(\text{Peter})(\text{they})$	$=$	$\text{love}(\text{Peter})(\mathbf{they})$

<u>Exercises</u>

Reduce the follwing expressions as much as possible by λ-conversion and provide their type.

(19)
a. $\lambda x[M(x)](j)$
b. $\lambda x \lambda y[P(y)(x)](a)(b)$
c. $\lambda x[\lambda y[P(x)(y)](a)](b)$
d. $\lambda x \lambda P[P(x)](j)(M)$
e. $\lambda x[\lambda y[P(x)(y)](a)]$
f. $\lambda P[P(j)](\lambda x[M(x)])$
g $\lambda x[\lambda y[P(x)(y)]](a)$
h. $\lambda P \lambda x[P(x)]\ (\lambda z[z=a])$
i. $\lambda P \lambda x[P(a)(b)]\ (\lambda x \lambda c[Q(x)(c)])$
j $\lambda x \lambda P \lambda y[P(x) \wedge \neg x=y](a)(\lambda z[Q(z)(a)])$

<u>Free vs. Bound variables</u>

The two computations below, which differ only in the variable converted into the λ-term, yield different results: Only the first one preserves meaning.

(20) $\exists y[x \neq y]$ Assume: $x = z$ (or: let x and z be bound e.g. by $\lambda P \lambda w[P(w)]$)

(21)
a. $\lambda x[\exists y[x \neq y]]$ *Abstraction over x*
b. $\lambda x[\exists y[x \neq y]](z)$ *Conversion of variable which is free in the scope of the λ-operator*
c. $\exists y[z \neq y] \quad \Leftrightarrow \quad \exists y[x \neq y]$

(22)
a. $\lambda x[\exists y[x \neq y]]$ *Abstraction over x*
b. $\lambda x[\exists y[x \neq y]](y)$ *Conversion of variable which is bound in scope of λ (bound by ∃)*
c. $\exists y[y \neq y] \quad \not\Leftrightarrow \quad \exists y[x \neq y]$

☞ Thus, in working with the λ-calculus it is important to keep in mind the theorem in (23), which prohibits accidental binding of free variables.

(23) *Theorem:* Free variables must not get accidentally bound by λ-conversion!

8.3. APPLYING THE λ-CALCULUS

The λ-notation provides a less awkward way of calculating the meaning of sentences, which expressed the compositional nature of the translation procedure in a transparent way.

(24) Bill read the book

(25) $\llbracket \textbf{read} \rrbracket \;=\; \lambda x\lambda y[\mathrm{read}(x)(y)]$
$\llbracket \textbf{book} \rrbracket \;=\; \lambda x[\mathrm{book}(x)]$
$\llbracket \textbf{the} \rrbracket \;=\; \lambda P[\text{the unique a such that } P(a) = 1]$

Presupposition: Defined for any $P \in D_{<e,t>}$, iff there is exactly one $a \in D_e$ such that $P(a)=1$

Sample Derivation

(26)

1. $\llbracket [_{IP}\ [_{NP}\ \textbf{Bill}]\ [_{VP}\ \textbf{read}\ [_{NP}\ \textbf{the book}]]] \rrbracket \;=\; 1$ iff
2. $\llbracket [_{VP}\ \textbf{read}\ [_{NP}\ \textbf{the book}]] \rrbracket (\llbracket [_{NP}\ \textbf{Bill}] \rrbracket) \;=\; 1$ iff *FA*
3. $\llbracket [_{VP}\ \textbf{read}] \rrbracket (\llbracket [_{NP}\ \textbf{the book}] \rrbracket) (\llbracket [_{NP}\ \textbf{Bill}] \rrbracket) \;=\; 1$ iff *FA*
4. $\llbracket \textbf{read} \rrbracket (\llbracket \textbf{the book} \rrbracket) (\llbracket \textbf{Bill} \rrbracket) \;=\; 1$ iff *Non-branching nodes*
5. $\lambda \mathrm{x}\lambda \mathrm{y}[\mathrm{read(x)(y)}](\lambda P[\text{the unique a such that } P(\mathrm{a}) = 1](\lambda x[book(x)]))(\mathrm{Bill}) = 1$ iff
6. $\lambda \mathrm{x}\lambda \mathrm{y}[\mathrm{read(x)(y)}](\text{the unique a such that } \lambda x[\mathrm{book}(x)])(a))(\mathrm{Bill}) \;=\; 1$ iff
7. $\lambda x\lambda \mathrm{y}[\mathrm{read}(x)\mathrm{(y)}](\text{the unique a such that book(a)})(\mathrm{Bill}) \;=\; 1$ iff
8. $\lambda y[\text{read(the unique a such that book(a))}(y)](Bill) \;=\; 1$ iff
9. $\text{read(the unique a such that book(a))(Bill)} \;=\; 1$

Lexical insertion applies in step 5, λ-conversion in steps 6-9. The λ-terms to be converted in the next step are marked in italics).

8.4. RELATIVE CLAUSE REVISITED

Predicate Modification can be defined as follows in the λ-notation:

PM. Predicate Modification

If α has the form XP, and $\llbracket \beta \rrbracket, \llbracket \gamma \rrbracket \in D_{<e,t>}$

$\beta \quad \gamma$ *λ-notation*

then $\llbracket \alpha \rrbracket = \lambda x[\llbracket \beta \rrbracket(x) \wedge \llbracket \gamma \rrbracket(x)]$

● Relative clauses contain an empty operator movement chain (Chomsky 1977):

(27) the book $[_{CP}\ \mathrm{OP}_i$ which Bill read $t_i]$

Assume that the operator is interpreted as a λ-binder and that operator movement leads to λ-abstraction (for more on the details of this strategy see below).

(28) $[\![\mathbf{OP}_i\ \alpha]\!] = \lambda x[[\![\alpha]\!]]$

(29) 1. $[\![[_{CP}$ **which Bill read** $\mathbf{t_i}]]\!]$ =
$= \lambda x\lambda y[read(x)(y)](x)(Bill) =$ *Lexical Insertion*
$= \lambda y[read(x)(y)](Bill) =$ *λ-Conversion*
$= read(x)(Bill) =$ *λ-Conversion*

2. $[\![[_{CP}\ \mathbf{OP_i}$ **which Bill read** $\mathbf{t_i}]]\!]$ =
$= \lambda x[read(x)(Bill)]$ *λ-Abstraction*

3. $[\![[_{NP}$ **book** $[_{CP}\ \mathbf{OP_i}$ **which Bill read** $\mathbf{t_i}]]]\!]$ =
$= \lambda x[\lambda y[book(y)](x) = 1 \wedge \lambda z[read(z)(Bill)](x) = 1] =$ *PM*
$= \lambda x[book(x) = 1 \wedge read(x)(Bill) = 1]$ *λ-Conversion*

8.5. A FURTHER PROBLEM FOR RELATIVE CLAUSES

Pronouns in the scope (roughly: the c-command domain) of a quantifier can be interpreted either deictically as referring expressions (for more on that see below), or as variables whose interpretation covaries with the interpretation of the quantificational term:

(30) No driver found his keys
a. For no x, s.t. x is a driver, x found y's keys : referential reading
b. For no x, s.t. x is a driver, x found x's keys : bound variable reading

It is easy to see that the two readings are independent, and that pronouns are therefore ambiguous between these two interpretations:

❒ Assume a scenario where Jeff and Martin are the drivers, and none of them found their own keys, but they found Sam's keys. Then, (30) comes out as false in the referential reading (30)a, but as true in the bound interpretation (30)b.

❒ Assume alternatively that while once again Jeff and Martin are the drivers, Jeff found his own keys, but nothing else happened in this scenario. In this situation, (30) is judged as true in the referential interpretation, and as false if the pronoun is construed as a bound variable.

● On the most direct hypothesis bound variable pronouns are translated in the same way as traces, that is they denote variables (for expository convenience, it will be assumed again that there is only a single variable):

(31) $[\![\mathbf{her}]\!] = x$

Contexts which combine traces inside relative clauses and pronouns such as (32) pose now the following problem

(32) Everybody knows a woman who liked him

If both the trace and the pronoun are translated as variable ‘x’, the resulting reading does not match the intuitive truth-conditions (lets abstract away from the detailed analysis of the quantifier, and assume that it can be compositionally integrated, concentrating for present purposes on the NP-head of the relative clause):

(33)	1.	⟦**OP t liked him**⟧	=	λx[⟦**t liked him**⟧]	*OP-Rule*
	2.	λx[⟦**t liked him**⟧]	=	λx[⟦**liked**⟧(⟦**him**⟧)(⟦**t**⟧)]	*FA*
✘	3.	λx[⟦**liked**⟧(⟦**him**⟧)(⟦**t**⟧)]	=	λx[⟦**liked**⟧(x)(x)]	*Lexical insertion*

Consider now the meaning of the last line above, which denotes a reflexive predicate. However, clearly, (32) does not involve reflexivization.

The problem arising in this context is that traces and pronouns bear indices, which have to be taken into account in the interpretation. For this reason, the semantic rules have to include a rule which translates variables according to the values which is assigned to them.

9. PRONOUNS

(1) a. Ann slept
b. She_7 slept

Pronouns are interpreted as variables, and not as constants. This means that while *Ann* can be given a meaningful interpretation in contexts where there is an individual called 'Ann', pronouns lack reference of their own. Rather, they are interpreted by *assigning* them a referent. This assignment of reference is rotationally expressed in terms of the index (subscripted natural number) on the pronoun. That is, it is the index and not the lexical material which contributes the interpretation of the pronoun.

(2) *Assumption:* Only the index on a pronoun is interpreted.

(3) For any $i \in \mathbb{N}$ and any $\alpha \in D_{pronoun}$: $[\![\alpha_n]\!] = [\![n]\!]$

> *Notation Convention:* Syntacticians generally use lower case letters as indices while semanticists use both letters and numbers (subject to personal preference). Nothing bears on this matter, though, as long as one remains consistent.)

9.1. TYPES OF PRONOUNS

Traditionally, pronouns are grouped together in two classes, depending upon whether they get their interpretation from the utterance situation (e.g. by pointing) or from the linguistic context. If the reference of a pronoun is fixed by the utterance context, as in (3), it is called a *deictic* pronoun.

(4) a. She_{12} lost the keys
b. He_5 is the murderer

But pronouns can also be used *anaphorically***,** in case of which the referent is determined by some other linguistic expression in the discourse. For instance, while the first pronoun in (4)b is traditionally said to be deictic, because there is no linguistic antecedent, the second one is ambiguous between a deictic (index 11) and an anaphoric (index 6) interpretation:

(5) She_6 lost $her_{6/11}$ keys

Both types of pronouns introduced above are *referential* in the sense that one interprets them as standing in for some individual. If the are interpreted as coreferential with some linguistic antecedent, this antecedent therefore also has to refer to some individual. They differ in this respect from ***bound variable pronouns*** ,which are interpreted as variables.

> NB: The latter property will relate pronouns to the relative clause puzzle!

(6) a. $Nobody_3$ lost his_3 keys
b. $\neg\exists$x[person(x) $\wedge$ x_lost_**x**'s_keys]

9.2. ASSIGNMENT FUNCTIONS

Pronouns are assigned their reference by ***assignment functions*** - usually written ‘g’. An assignment function is a function from indices to individuals which yields as its value the meaning of that index. For instance, if one picks the specific assignment function g below, the pronoun *she*$_7$ will refer Mary:

(7) Situation 1: She$_7$ arrived
$g(7) = \text{Mary}$
$[\![\text{She}_7 \text{ arrived}]\!]^g \Leftrightarrow [\![\text{Mary arrived}]\!]^g$

There are two kinds of individual in D_e: constants and variables. Only the meaning of the latter is dependent upon an assignment function. Thus, the two types of terms are assigned denotations in the following way:

(8) $[\![\alpha]\!]^g = \alpha$ if α is a constants
$[\![\alpha]\!]^g = g(\alpha)$ if α is a variable

The choice of assignment function depends to the context. That is, if in a situation the speaker has been waiting for Mary, and then utters *She arrived* once she has reached her destination, the sentence is most likely[11] interpreted by using the assignment function g above. If however the speaker is expecting Susan, as in Situation 2, he/she will most likely assign to 7 the value Susan, instead of Mary:

(9) Situation 2: She$_7$ arrived
$g'(7) = \text{Susan}$
$[\![\text{She}_7 \text{ arrived}]\!]^g \Leftrightarrow [\![\text{Susan arrived}]\!]^g$

Thus, context dependence can be achieved by using assignment functions which differ in the individual they assign to certain indices.

Variables in Linguistis and in Logic

Semantic (individual) variables come in two guises: traces and pronouns. In linguistics, they are not distinguished in the same way as in logic, where different variables are assigned to different letters; instead, they are kept apart by subscripts. Thus, the ‘variable’ part is in fact the index on the trace/pronoun. In order to keep in line with this tradition, the assignment function g will be taken to apply to the index, and not to the trace itself:

(10) $[\![\alpha_i]\!]^g = g(i)$ if is a variable

For instance, the denotation of **Sally** is Sally, while the denotation of the pronoun **he**$_1$ is Tom under assignment g, but Sally under assignment g’ below:

[11]The qualification ‘most likely’ is necessary, because which assignment function is finally chosen is entirely determined by the context, i.e. extralinguistic factors such as the design of the situation, speakers’ intentions, wishes and hopes, etc... The actual choice of the assignment functions accordingly varies with the context.

(11) a. $g = \begin{bmatrix} 1 & \rightarrow & \text{Tom} \\ 2 & \rightarrow & \text{Sally} \end{bmatrix}$

$g' = \begin{bmatrix} 1 & \rightarrow & \text{Sally} \\ 2 & \rightarrow & \text{Mary} \end{bmatrix}$

b. $[\![\text{he}_1]\!]^g = g(1) =$ Tom

$[\![\text{he}_1]\!]^{g'} = g'(1) =$ Sally

Question: How can it be ensured that antecedent and pronoun matchin gender features?

9.3. Relative clauses & Assignment Functions

What else is needed in order to employ assignment functions in the interpretation of relative clauses? Recall that the problem was that in example (32), repeated below, translating both the trace and the pronoun as variable 'x', the resulting reading yielded a reflexive predicate:

(12) Everybody knows a woman who liked him (= (32))

(13) a. $[\![\textbf{OP t liked him}]\!] = \lambda x[[\![\textbf{t liked him}]\!]]$ *OP-Rule*

b. $\lambda x[[\![\textbf{t liked him}]\!]] = \lambda x[[\![\textbf{liked}]\!]([\![\textbf{him}]\!])([\![\textbf{t}]\!])]$ *FA*

✘ c. $\lambda x[[\![\textbf{liked}]\!]([\![\textbf{him}]\!])([\![\textbf{t}]\!])] = \lambda x[[\![\textbf{liked}]\!](x)(x)]$ *Lexicon pronoun & trace*

But section 9.2. introduced a way to distinguish the pronoun and the trace in terms of assignment functions. Thus, one gains:

(14) a. $[\![\textbf{t}_1\textbf{ liked him}_6]\!]^g = [\![\textbf{liked}]\!]^g([\![\textbf{him}_6]\!]^g)([\![\textbf{t}_1]\!]^g)$ *FA*

b. $[\![\textbf{liked}]\!]^g([\![\textbf{him}_6]\!]^g)([\![\textbf{t}_1]\!]^g) = [\![\textbf{liked}]\!]^g(g(6))\ (g(1))$ *Interpreting pronoun & t*

But it has not been ensured yet that the variable t_1, the trace of the empty operator, co-varies with its antecedent. That is, the final representation should look as follows:

(15) $\forall x[\text{human}(x) \rightarrow \exists y[\text{woman}\ (y) \wedge \text{know}(y)(x) \wedge \text{like}(x)(y)]]$

However, if one applies the lexical entry for the empty operator employed so far, the result is (17):

(16) $[\![\textbf{OP}\ \alpha]\!] = \lambda x[[\![\alpha]\!]]$

(17) a. $[\![\textbf{OP}_1\ \textbf{t}_1\textbf{ liked him}_6]\!] =$

b. $= \lambda x[[\![\textbf{t}_1\textbf{ liked him}_6]\!]^g] =$

c. $= \lambda x[[\![\textbf{liked}]\!](g(6))(g(1))]$

and this is certainly not the correct result. Switching for ease of exposition to set talk, (17) denotes the set of all individuals in the domain if $[\![\textbf{t}_1\textbf{ liked him}_6]\!]^g = 1$, - that is if the individual denoted by g(1) likes the individual denoted by g(6) - and the empty set otherwise.

<u>Exercise</u>

Design a scenario and an assignment function under which (17) denotes the predicate which is true of all individuals.

Desiderata for Assignment Functions

The deliberations above support two conclusions:

❒ First, assignment functions have to be designed in such a way that they do not only return individual constants as their values, but also range over variables.

(18) Let g(1) = x
Let g(6) = y
$[\![\mathbf{t_1}]\!]$ = g(1) = x
$[\![\mathbf{him_6}]\!]$ = g(6) = y

(19) 1. $[\![\mathbf{OP_1\ t_1\ liked\ him_6}]\!]$ = $\lambda x[[\![\mathbf{liked}]\!](g(6))(g(1))]$
2. $\lambda x[[\![\mathbf{liked}]\!](g(6))(g(1))]$ = $\lambda x[[\![\mathbf{liked}]\!](y)(x)$

❒ Second, it must be ensured that they introduce the *correct* variable! While the representation above would indeed do, as it lets the λ-operator bind the subject position of **like**, the following translation, where g(1) yields z instead of x has e.g. to be excluded:

(20) $\lambda x[[\![\mathbf{liked}]\!](g(6))(g(1))]$ = $\lambda x[[\![\mathbf{liked}]\!](y)(z)]$ ✘

Goal: Define an operation which makes it possible to let variables in the domain of the λ-operator co-vary with the index of the trace.

9.4. MODIFIED ASSIGNMENT FUNCTIONS

In order to connect the index of the trace and the λ-term, it turns out to be useful to define the notation of a modified variable assignment.[12]

(21) $g^{[x/a]}$ is the ***modified variable assignment*** g' which possibly differs from g only in that it assigns a to x.

Example

(22) Let g(3) = Sally
g(5) = Tom

a. $[\![t_3]\!]^{g}$ = g(3) = Sally
b. $[\![t_3]\!]^{g[3/Mary]}$ = $g^{[3/Mary]}(3)$ = Mary
c. $[\![t_5]\!]^{g}$ = g(5) = Tom
d. $[\![t_5]\!]^{g[3/Mary]}$ = $g^{[3/Mary]}(5)$ = Tom
e. $[\![t_5]\!]^{g[3/Mary,\ 5/Mary]}$ = $g^{[3/Mary,\ 5/Mary]}(5)$ = Mary

[12] Interpreting quantified formulas by temporary assignments was first suggested by Alfred Tarski in the 1940ies. Using this method, Tarski was able to design the first *compositional* treatment of first order logic (predicate calculus).

○ Why is there the qualification 'possibly' in the definition of modified assignment function? Because g and g' are identical if g(x) = a. For instance,

(23) Let $g(3) \quad = \quad$ Mary
Then $[\![t_3]\!]^{g[3/Mary]} \quad = \quad g^{[3/Mary]}(3) \quad = \quad$ Mary $= g(3)$

○ The notations in (24)a and (24)b are equivalent:

(24) Let $g(1) = $ Tom
$g(2) = $ Sally

a. $g = \begin{bmatrix} 1 & \rightarrow & \text{Tom} \\ 2 & \rightarrow & \text{Sally} \end{bmatrix}$

b. $g^{[2/Mary]} = \begin{bmatrix} 1 & \rightarrow & \text{Tom} \\ 2 & \rightarrow & \text{Sally} \end{bmatrix} [2 \rightarrow \text{Mary}]$

A remark on *notational conventions:* The literature employs a confounding number of different ways to graphically encode modified assignment functions. It is possible to find at least the following notations, which all mean the same.

(25)
a. $g^{[2/Mary]}$ (e.g. GAMUT [modulo type face], and present manuscript)
b. $g^{[2 \rightarrow Mary]}$ (e.g. Heim & Kratzer 1998)
c. $g^{[2:= Mary]}$ (e.g. Barendregt 1984)
d. g^{Mary}_{2} (e.g. Dowty, Wall & Peters 1981)
e. $g^{[Mary/2]}$ (e.g. in classic Montague grammar, mathematical logic)

9.5. Relative Clauses revisited

Modified assignment functions provide an elegant way to ensure that the index on the trace of the empty operator is assigned the same value as the variable which the λ-operator is defined for.

(26) *Assumptions*
- ○ Empty operator is interpreted as a λ-abstractor over the individual variable it binds.
- ● Only the index on the trace is interpreted (just like with pronouns).

PA. Predicate (Lambda-) Abstraction (final version)

For any g and any $n \in \mathbb{N}$, if α has the form $[_{XP}\ OP_n\ \gamma]$, then $[\![\alpha]\!]^g = \lambda x[[\![\gamma]\!]^{g[n/x]}]$

Sample Derivation

(27) movie that Bill had rented

(28) a. $[\![[_{CP}\ \mathbf{OP_1\ \ that\ Bill\ had\ rented\ t_1}]]\!]^g$ =

= $\lambda x[[\![\mathbf{that\ Bill\ had\ rented\ t_1}]\!]^{g[1/x]}]$ = *Predicate Abstraction*

= $\lambda x[[\![\mathbf{Bill\_had\_rented}]\!]^{g[1/x]}\ ([\![\mathbf{t_i}]\!]^{g[1/x]})]$ =

= $\lambda x[\text{Bill\_had\_rented}\ ([\![\mathbf{t_1}]\!]^{g[1/x]})]$ =

= $\lambda x[\text{Bill\_had\_rented}\ (g^{[1/x]}([\![\mathbf{t_1}]\!]))]$=

☞ = $\lambda x[\text{Bill\_had\_rented}\ (g^{[1/x]}(1))]$ =

✓ = $\lambda x[\text{Bill\_had\_rented}\ (x)]$

b. $[\![\mathbf{movie\ [_{CP}\ OP_1\ that\ Bill\ had\ rented\ t_1}]]\!]^g$ =

= $\lambda x[\text{movie}(x)]\ \lambda x[\text{Bill\_had\_rented}(x)]$ = *Predicate Modification*

= $\lambda x[\text{movie}(x) \wedge \text{Bill\_ had\_rented}(x)]$

APPENDIX: FURTHER APPLICATIONS OF Λ-CALCULUS

Conjunction

(29) a. Sam sings and Mary smokes
b. $[\![\mathbf{and_1}]\!]$ = $\lambda p_{<t>}\ \lambda q_{<t>}\ [q = 1 \wedge q = 1]$

(30) a. Sam sings and smokes
b. $[\![\mathbf{and_2}]\!]$ = $\lambda P_{<e,t>}\ \lambda Q_{<e,t>}\ \lambda x[P(x) = 1 \wedge Q(x) = 1]$

(31) a. Sam bought and ate the pie
b. $[\![\mathbf{and_3}]\!]$ = $\lambda P_{<e,<e,t>>}\ \lambda Q_{<e,<e,t>>}\ \lambda x\lambda y[P(x)(y) = 1 \wedge Q(x)(y) = 1]$

(32) a. Sam and Mary sing
b. $[\![\mathbf{and_4}]\!]$ = $\lambda x\lambda y\lambda P[P(x) = 1 \wedge P(y) = 1]$

Passive

(33) a. Syntax: The book was written by Arp
b. Semantics Input: $[_{IP}$ The book $[_{VP}\ [_{V^\circ}\text{write}_{Pass}\ ]$ Arp]]

(34) $[\![\mathbf{write_{active}}\ ]\!]$ = $\lambda x\lambda y[\mathbf{write}(x)(y)]$
$[\![\mathbf{written}]\!]$ = $[\![\mathbf{write}_{passive}]\!]$ =
= $\lambda x\lambda y[\mathbf{write}_{active}\ (y)(x)]$

(35) $[\![\mathbf{Passive}]\!]$ = $\lambda P_{<e,<e,t>>}\ \lambda x\lambda y[P(y)(x)]$
$\lambda P\lambda x\lambda y[P(y)(x)](\lambda w\lambda z[\mathbf{write}_{active}\ (w)(z)])$ = *λ-Conversion*
= $\lambda x\lambda y[\lambda w\lambda z[\mathbf{write}_{active}(w)(z)](y)(x)]$ = *λ-Conversion*
= $\lambda x\lambda y[\lambda z[\mathbf{write}_{active}(y)(z)](x)]$ = *λ-Conversion*
= $\lambda x\lambda y[\mathbf{write}_{active}(y)(x)]$

Reflexives

Lexical/semantics approaches towards reflexivization[13]: reflexive pronouns turn non-reflexive predicates into reflexive ones:

(36) Syntax: $[_{IP}$ Sam $[_{VP}$ saw herself$]]$
a. LF: $[_{IP}$ Sam $[_{VP}$ $[_{V°}$ self$_i$ $[_{V°}$saw$]]$ t_i $]]$
b. Semantics: $[_{IP}$ Sam $[_{VP}$ $[_{V°}$ self $[_{V°}$saw$]]]]$

Exercise

Assume the lexical entry for the semantic translation of **self**, assuming that for any two-place predicate P, **self-P** is the diagonalization of P. Compute the meaning of *Sam saw herself.* Does the analysis carry over to *I showed Bill to himself*? If not, what would an LF have to look like which derives the correct interpretation?

(37) $\lambda P_{<e,<e,t>>} \lambda x[P(x)(x)]$

Diagonalization: the function which assigns each individual in D to itself (f(x) = x):

(38)

x *f(x)* =	1	2	3	4
1	○			
2		○		
3			○	
4				○

[13]See e.g. Bach, Emmon and Barbara H. Partee. 1980. Anaphora and Semantic Structure. In: J. Kreiman and A. E. Ojeda (eds.), *Papers from the Parasession on Pronouns and Anaphora*. Chicago: University of Chicago Press.
Keenan, Edward. 1988. On semantics and the binding theory. In J. Hawkins (ed.), *Explaining Language Universals*. Oxford: Blackwell, pp 104–155.
Reinhart, Tanya and Eric Reuland. 1993. Reflexivity. *Linguistic Inquiry* **24.**4: 657-720.
Szabolcsi, Anna. 1987. Bound variables in syntax. (Are there any?). In J. Groenendijk and M. Stockhof (eds.), *Proceedings of the 6th Amsterdam Colloquium.* Amsterdam: University of Amsterdam, pp. 331-353. Available for download at: http://semanticsarchive.net/Archive/GQyODUwZ/

10. QUANTIFICATION

NEXT:
- ○ Quantifiers vs. referring expressions.
- ● Interpretation: a first approximation
- ○ Compositional interpretation of quantifiers (λ-notation)
- ● Quantifiers in object position and inside NPs: Quantifier raising and its interpretation
- [○ Logical properties of quantifiers (existential constructions, extraposition, ...)]
- [● The semantics of quantification (Tarski's temporary assignments)]

10.1. NAMES VS. QUANTIFIERS

The object NP in (1)a is a name, while the object of (1)b is represented by a ***quantifier****:*

(1) a. Sam read *Ulysses/Kratylos/*the telephone book of Uppsala.
b. Sam read every book/some book/no book/most books..

(2) a. Names: *Ulysses*, *Kratylos*, *Fear and Loathing in Las Vegas*, *Gone with the Wind*, ...
b. Quantifiers: every book, some book, no book, more than two books, half of the books, most books, ...

Quantifiers systematically differ in their referential properties from names: unlike names, quantifiers do not refer to individuals. Thus, they cannot be treated as symbols of type e.

1. Lack of Reference

Names refer to an individual in the world. The expression *Sam* refers e.g. to the person with the name *Sam*, and *Ann* to the woman which happens to bear the name *Ann*. Therefore, we know that sentence (3) is true in all those situations in which Sam likes Ann, and false otherwise:

(3) Sam likes Ann.

Quantifiers on the other side fail to refer to individuals. This becomes especially evident with negative existential quantifiers ($\neg\exists x\Phi$ or $\forall x\neg\Phi$):

(4) a. No dog likes Ann. $\neg\exists x[dog(x) \wedge likes(Ann)(x)]$
b. Ann likes no dog. $\neg\exists x[dog(x) \wedge likes(x)(Ann)]$

On this conception, quantifiers operate on sets, which in turn are made up of individuals. A quantifier such as *no dog* in (4) relates e.g. the set of dogs with the set of individual who like Ann or who are liked by Ann.

2. Law of the excluded middle

(5) *Question:* Are the following two statements true or false?

(6) Sam is French or Sam is not French
(7) Everybody is French or everybody is not French.

Generally, the law of the excluded middle - a law of 2-valued logic - holds that for any p, it is always true that 'p or not p' holds. Assume that p is valued as a VP-denotation, as in (8)a below. Combining this VP with a name in subject position results in a tautology. (8)b is true, irrespective whether Sam is a p (i.e. French) in any given situation:

(8) a. Let p = be French
$\vDash$ $p \vee \neg p$ $\rightarrow$ be French or be not French $(= D_e)$
b. Sam is French or Sam is not French

(9) *Law of excluded middle*
a. General: $\vDash p \vee \neg p$
b. Specific: $\vDash \forall x \in D_e\ \forall p \in D_{<e,t>}[x \in p \vee x \in \neg p]$

NOTATIONAL CONVENTION: '$\vDash$' denotes semantic entailment.

(10) a. $A \vDash B$ *A semantically entails B*
b. $\vDash A$ *A is a tautology*
c. $\nvDash A$ *A is a contradiction*

If any antecedent entails A, A is universally valid or a tautology. If no antecedent entails A, that is if A is false whatever the choice of the antecedent, A is a contradiction. If A is neither a tautology or a contradiction, A is a *contingency.* That is, the truth of A is contingent on the choice of the antecedent: for some, A will come out as true, and for others false.

What is important for present purposes is that the law of the excluded middle only applies to referring expressions (names), but does not hold for quantifiers: While (8)b is a tautology, (11)b, in which the subject is a quantifier instead of a name, is not. The truth of (11)b is contingent upon the way the world is structured. It holds only if either everybody adopts French citizenship or if all French become e.g. Polish citizens.

(11) a. Let p = be French
b. Everybody is French or everybody is not French
c. $\neg\vDash\ \forall QP \forall_p \in D_{<e,t>}\ [QP \in p \vee QP \in \neg p]$

It follows now that quantifiers cannot be treated as referring expressions.

Further example: Here is a further example involving the law of the excluded middle: The following sentence once again is a tautology, it is true independent of the actual length of the book *Ulysses*.

(12) *Ulysses* is more than 100 pages long or *Ulysses* is less than 1,000 pages long.

Substituting the name by a quantifier leads to a sentence which is no longer a tautology. In the actual world, (13) is e.g. evaluated as false. This is so because there are some books less than 100 pages long - falsifying the first sentence in the disjunction - and there are also some books which are longer than 1,000 pages, falsifying the second sentence.[14]

(13) Every book is more than 100 pages long or every book is less than 1000 pages long.

Law of contradiction

Consider finally the a pair of statements below, which differ only w.r.t. the subject position.

(14) Ulysses is boring and Ulysses is not boring.
(15) Some book is boring and some book is not boring.

Sentence (14) can never be true, irrespective of the actual situation. That is, (14) is a contradiction, as are all statements of the form 'p and not p'. Again, sentences of the same shape in which the name has been substituted by a quantifier behave differently: Unlike (14), (15) can be used as a truthful description of some situation.

10.2. Interpreting Quantified Sentences

It was pointed out that set notation can be substituted for function notation and v.v. For instance, noun phrases such as $[_{NP}$ **book**] denote characteristic functions of sets of individuals (*λx[book(x) =1]*). The set theoretic notation offers an intuitive and transparent way to conceptualize the effects of natural language quantification.

Recall that given the set theoretic translation of predicates, function application is reinterpreted as set membership:

(16) ⟦**dog**⟧ = {Rover, Sue}
⟦**sleep**⟧ = {Rover, Sue, Pat}

(17) ⟦**Sam sleeps**⟧ = Rover ∈ ⟦**sleep**⟧

Consider now the quantification structures as in (18)a:

(18) a. Every dog sleeps
b. No dog sleeps

Intuitively, quantifiers establish relations between sets: (18)aa means that the set of dogs is a subset of the set of individuals who sleep. And (18)ab means that the intersection of the set of dogs and the set of individuals who sleep is empty[15].

(19) a. $\{x|x \text{ is a dog}\} \subseteq \{x|x \text{ sleeps}\}$ (= (18)aa)
b. $\{x|x \text{ is a dog}\} \cap \{x|x \text{ sleeps}\} = \{\} = \varnothing$ (= (18)ab)

---

[14](12) and (13) can be reduced to the general format 'p ∧ ¬p' by the following inference: Assume that p stands for the statement 'Ulysses is more than 100 pages long'. Then 'not p' is the statement 'It is not the case that Ulysses is more than 100 pages long'. From this, one can infer the truth of the statement q = 'Ulysses is less than 1000 pages long'. This is so because everything which is 100 or less pages long is also less than 1,000 pages long. One can now substitute q for 'not p', resulting in (1), (12).

[15]Euler (1768) was the first to employ Leibniz's device of illustrating logical relations by geometrical relations to model three of the four Aristotelian forms of statements (*all*, *no* and *some*).

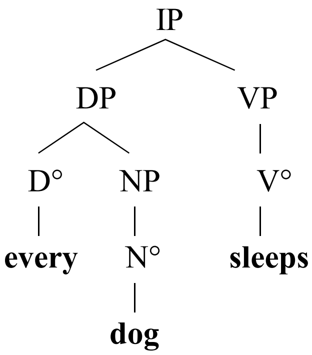
10.2.1. Syncategorematic Translation

As a first approximation towards a semantics for quantified sentences, assume the following (syncategorematic; see below) lexical entries for *every* and *no*:

(20) $[\![$**every** $\alpha\ \beta]\!] = 1$ iff $[\![\alpha]\!] \subseteq [\![\beta]\!]$
$[\![$**no** $\alpha\ \beta]\!] = 1$ iff $[\![\alpha]\!] \cap [\![\beta]\!] = \varnothing$

On this view, one just needs to plug in the denotation of the quantifier to gain the correct result:

(21)



(22) $[\![$**Every dog sleeps**$]\!] =$ 1 iff $[\![$**dog**$]\!] \subseteq [\![$**sleeps**$]\!]$

However, this method suffers from serious flaws. Notice that the definitions for *every* and *no* in (20) differ from all other lexical entries that we have encountered so far. Usually, the lexical entry of a category α was of the form $[\![\alpha]\!] =$ ____, while in (20), the denotation brackets also include two variables, which are instantiated by the common noun (*dogs*) and the VP (*sleep*). Strictly speaking, we do not assign a meaning to *every*, but to the (IP) node resulting from applying *every* to a common noun and a VP. Such a translation in which the lexical item is assigned a meaning only in context with other nodes is referred to as **SYNCATEGOREMATIC** treatment. Clearly, syncategorematic rules violate the spirit of the principle of compositionality, in that they obfuscate the interpretation of the internal structure of a given constituent. This is undesirable as it fails to account for the generative capacity of natural language, i.e. the observation that a finite (small) number of recursive rules derives the infinite set of all well-formed sentences.

10.2.2. Categorematic Translation

A first version of categorematic lexical entries for the universal quantifier and the negative existential can be given as below:

(23) $[\![$**every**$]\!](A)(B) = 1$ iff $A \subseteq B$
$[\![$**no**$]\!]\ (A)(B) = 1$ iff $A \cap B = \varnothing$

Furthermore, once the lexical entry is changed slightly such that the two argument positions (A and B) are abstracted over by λ-abstraction, it becomes possible to interpret quantified clauses compositionally. Thus, unlike in the categorematic analysis, every node is assigned a meaning:

(24) $[\![$**every**$]\!] = \lambda P \lambda Q[P \subseteq Q]$

Sample Derivation

(25)

```
              IP
            /    \
          DP      VP
         /  \      |
       D°    NP    V°
       |     |     |
     every   N°   sleeps
             |
            dog
```

(26)	1. ⟦**Every dog sleeps**⟧	=	1 iff		
	2. ⟦**every**⟧(⟦**dog**⟧)(⟦**sleeps**⟧)	=	1 iff	*FA*
	3. λPλQ[P ⊆ Q](⟦**dog**⟧)(⟦**sleeps**⟧)	=	1 iff	*Lexicon*
	4. λQ[⟦**dog**⟧ ⊆ Q] (⟦**sleeps**⟧)	=	1 iff	*λ-Conversion*
	5. ⟦**dog**⟧ ⊆ ⟦**sleeps**⟧			*λ-Conversion*

Note that the denotation of the subject DP applies to the VP-denotation, and not vice versa (as was the case in the examples encountered so far)! This follows directly from the theory of types (see below).

Quantifiers measure how many As are Bs in Q(A)(B):

(27) a. Some dogs sleep
b. Most dogs sleep
c. At least two dogs sleep

(28) a. ⟦**some**⟧ (A)(B) = 1 iff A ∩ B ≠ ∅
b. ⟦**most**⟧(A)(B) = 1 iff |A∩B| > ½ |A| or
⟦**most**⟧(A)(B) = 1 iff |A∩B| > |A - B|
c. ⟦**at least two**⟧(A)(B) = 1 iff |A∩B| ≥ 2

10.2.3.Translation into Predicate Logic

Although it is in principle possible to generalize the set theoretic analysis to other quantifiers, sets lack the expressive power to account for the fact that ordering plays a role in the interpretation of sentences (relations, ordered pairs, etc...). Thus, we revert to the notation in terms of Predicate Logic in what follows. The set theoretic notions of intersection, negation and union can also be expressed in terms of first order predicate logic (using ∀, ∃, ¬, ∧ and →).

(23) ⟦**every**⟧(A)(B) = 1 iff ∀x[A(x) → B(x)]
⟦**no**⟧ (A)(B) = 1 iff ¬∃x[A(x) ∧ B(x)]

Again, using the λ-notation yields the lexical entry which can be compositionally interpreted (in what follows, brackets will be omitted unless ambiguity arises).

(29) $[\![\textbf{every}]\!] = \lambda P\lambda Q[\forall[xP(x) \rightarrow Q(x)]]$
$[\![\textbf{no}]\!] = \lambda P\lambda Q[\neg\exists x[P(x) \wedge Q(x)]]$

It becomes now also possible to determine the logical type of quantifiers: They denote functions of type <<e,t>,<e,t>,t>, that is they are functions from characteristic functions of individuals to functions from functions from characteristic functions of individuals to truth values. Since (almost) all natural language quantifiers are of this type, such quantifiers are also called ***Generalized Quantifiers*** (following Mostowsky 1957; see Barwise and Cooper 1982).

Sample Derivation

At this point, everything which is needed in order to compute the compositional meaning of a sentence is in place. Observe that all lexical entries are treated as λ-terms, and that λ-conversion ensures that the variables are bound off correctly:

(30)
1. $[\![\textbf{Every dog sleeps}]\!] = 1$ iff
2. $[\![\textbf{every}]\!]([\![\textbf{dog}]\!])([\![\textbf{sleeps}]\!]) = 1$ iff — *FA*
3. $\lambda P\lambda Q[\forall x[P(x) \rightarrow Q(x)]]([\![\lambda y[\textbf{dog}(y)]\!])(\lambda z[\![\textbf{sleeps}(z)]\!]) = 1$ iff — *Lexicon*
4. $\lambda Q[\forall x\, [\![\lambda y[\textbf{dog}(y)]\!](x) \rightarrow Q(x)](\lambda z[\![\textbf{sleeps}(z)]\!]) = 1$ iff — *λ-Conversion*
5. $\forall x\, [\![\lambda y[\textbf{dog}(y)]\!](x) \rightarrow \lambda z[\![\textbf{sleeps}(z)]\!](x)] = 1$ iff — *λ-Conversion*
6. $\forall x\, [\![\textbf{dog}(x) \rightarrow \textbf{sleeps}(x)] = 1$ — *λ-Conversion*

Exercises

● Provide the predicate logic translation for the following sentences (ignoring tense, as usually):

(31)
a. Exactly one boy read the book which someone had recommended.
b. Sam does not like Sally but someone else.
c. They liked every movie except Titanic.
d. Birds lay eggs but not all mammals do not lay eggs.
e. Someone has stolen a book and has not returned it.

○ Design the lexical entries (λ-notation) for the three quantifiers used below:

(32)
a. At least one dog arrived
b. $\exists x[dog(x) \wedge arrive(x)]$

(33)
a. At least two dogs arrived.
b. $\exists x\exists y[x \neq y \wedge dog(x) \wedge dog(y) \wedge arrive(x) \wedge arrive(y)]$

(34)
a. The milkman arrived
b. $\exists x[milkman(x) \wedge \forall y[milkman(y) \rightarrow x = y] \wedge arrive(y)] =$
$= \exists x[\forall y[milkman(y) \leftrightarrow x = y] \wedge arrive(y)]$ (Russell 1905)

● Determine whether the statements below are true of false in the Model TL:

(35) MODEL TL
a. D = {m(ary), s(am), b(ill)}
b. ⟦**tired**⟧ = {m,s}
⟦**like**⟧ = {<m,s>, <m,b>, <b,s>, <s,s>}

(36) a. $\exists x \exists y \exists z[\text{like}(y)(x) \wedge \text{tired}(y) \wedge \text{like}(z)(x) \wedge \neg\text{tired}(z)]$
b. $\forall x[\neg\text{meet}(x)(x)]$
c. $\forall z[\text{meet}(z)(z) \leftrightarrow \neg\text{tired}(z)]$
d. $\exists x \exists y[\text{meet}(y)(x) \wedge \neg\text{tired}(x) \wedge \neg\text{tired}(y)]$
e. $\forall x[\text{meet}(x)(x) \rightarrow \exists y[\text{meet}(y)(x) \wedge \text{tired}(y)]]$
f. $\forall x[\text{tired}(x) \rightarrow \exists y[\text{meet}(y)(x)]]$

10.2.4. The limits of 1st order Predicate Logic

Is it now possible to find a suitable first order predicate logic interpretation for all quantifiers in natural language? The answer is negative, i.e. it works for some, but not for others. To illustrate, consider *most* (assume *most* means *more than half*):

(37) Most French hate Mozart

(38) a. ⟦**most**⟧ = $\lambda P \lambda Q[\text{most } x[P(x) \wedge Q(x)]]$
b. ⟦**most**⟧ = $\lambda P \lambda Q[\text{most } x[P(x) \rightarrow Q(x)]]$

The reason why it is impossible to translate determiners such as *most* (and *many, few*) into first order predicate logic becomes apparent once one consider the natural language paraphrases for (37) modeled after the lexical entries above:

(39) a. For most individuals: they are French and they hate Mozart
b. For most individuals: they are French if they hate Mozart

Neither version of (39)a not (39)b captures the meaning of (37)b correctly. While (39)a can be true only if there is a substantial inflation of French citizens, the truth conditions of (39)b require that for the larger part of the population, French citizenship is contingent upon specific musical tastes. But this is not what (37)b means.

This finding indicates that not all natural language quantifiers can be translated into first order predicate logic. Moreover, it shows that natural language employs so-called *restrictive quantification* (Mostowsky 1957, Rescher 1962, Geach 1972). Notice that the reason why (38) yields the wrong results is that the domain of *most* in the metalanguage translation is the whole universe (all individuals). But this gives now rise to truth condition which are too strong/weak. What is required is that *most* quantifies only over those individuals which are in the denotation of the first argument of the quantifier (*French*). Thus, the domain of quantification has to be restricted. A

Finally, two suitable lexical entries for *most* (in set notation; repeated from above):

(40) ⟦**most**⟧(A)(B) ⇔ $|A \cap B| > \frac{1}{2}|A|$ (Fintel 1994)
⟦**most**⟧(A)(B) ⇔ $|A \cap B| > |A - B|$ (Partee, ter Meulen, Wall 1993: 395)

10.3. QUANTIFIER RAISING

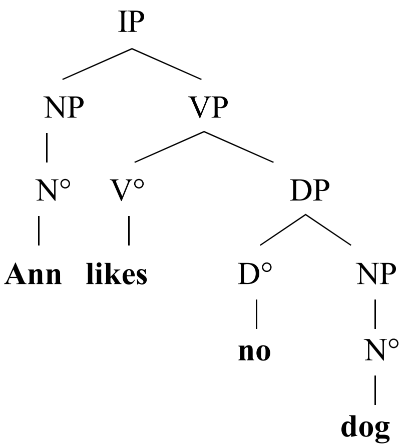
The treatment for quantifiers advocated above faces a serious problem: The lexical entries cover only cases in which the quantifier occupies the subject position of a sentence, as e.g. in (4)a:

(29) $[\![\textbf{every}]\!] = \lambda P\lambda Q[\forall x P(x) \rightarrow Q(x)]$
$[\![\textbf{no}]\!] = \lambda P\lambda Q[\neg\exists x P(x) \wedge Q(x)]$

(41)
a. No dog likes Ann. ✓ (= (4)a)
b. Ann likes no dog. ✘ (= (4)b)
c. She looked for it [$_{PP}$ in every corner] ✘
d. They invited [$_{NP}$ a student [$_{PP}$ from every European country]]. ✘

The lexical entry for *no* in (29) can however not be employed in the interpretation of (4)b, because in the tree for (4)b, there is no suitable NP and VP constituent which *no* could apply to in a compositional way. More precisely, $[\![\textbf{no dog}]\!]$ is of type <e,t>,t>, while $[\![\textbf{like}]\!]$ is of type <e,<e,t>>. Thus, they can neither be combined by Function application nor by Predicate Modification:

(42)



One possible solution to this problem consists in modulating the type of quantifiers or for verbs, such that they can be combined in a meaningful way (see. e.g. Heim and Kratzer 1998: 180f for discussion). A suitable lexical entry for *no* in object position would look as follows:

(43) $[\![\textbf{no}_{\textbf{Object}}]\!] = \lambda P_{<e,t>}\, \lambda Q_{<e,<e,t>>}\, \lambda x \neg\exists y[P(y) \wedge Q(y)(x)]$

This conception misses the insight though, that the meanings of QPs in different syntactic encironments are related to each other in a systematic way, resulting in a blow-up of the lexicon.

<u>Option I: Type Shifting</u>

Quantifiers (*no* in *no book*) are invariably treated as expressions of type <<e,t>,<<e,t>,t>, and so-called TYPE SHIFTING operators are introduced into the syntax, which change the type of the quantifier as required. For instance, the operator $\Uparrow_{\text{Object}}$ applies to QPs in object position, shifting them to expressions of type <<e,<e,t>,<e,t>>, which may combine with transitive verbs:

(44) $[\![\Uparrow_{\textbf{Object}}]\!] = \lambda \wp_{<<e,t>,t>}\, \lambda Q_{<e,<e,t>>}\, \lambda x[\wp(\lambda y[Q(y)(x)])]$

Sample Derivation

(45) a. They like no book

b. $[\![\text{They}_5\ [\text{like}\ [\Uparrow_{\text{Object}}\ [\text{no book}]]]]\!]^g =$

1. = $[\![\Uparrow_{\textbf{Object}}]\!]([\![\textbf{no book}]\!])([\![\textbf{like}]\!])([\![\textbf{they}_{\mathbf{5}}]\!]^g) =$

2. = $\lambda\wp\lambda Q\lambda x[\wp(\lambda y[Q(y)(x)])]$ $(\boldsymbol{\lambda P\neg\exists z[book(z) \wedge P(z)]})$ $(\lambda a\lambda b[like(a)(b)])(g(5)) =$

3. = $\lambda Q\, \lambda x[\lambda \mathbf{P}\neg\exists z[book(z) \wedge \mathbf{P}(z)](\boldsymbol{\lambda y[Q(y)(x)]})]$ $(\lambda a\lambda b[like(a)(b)])$ $(g(5)) =$

4. = $\lambda Q\lambda x[\neg\exists z[book(z) \wedge \lambda \mathbf{y}[Q(\mathbf{y})(x)](\mathbf{z})]]$ $(\lambda a\lambda b[like(a)(b)])$ $(g(5)) =$

5. = $\lambda \mathbf{Q}\lambda x[\neg\exists z[book(z) \wedge \mathbf{Q}(z)(x)]]$ $(\boldsymbol{\lambda a\lambda b[like(a)(b)]})$ $(g(5)) =$

6. = $\lambda x[\neg\exists z[book(z) \wedge \lambda \mathbf{a}\lambda b[like(\mathbf{a})(b)](\mathbf{z})(x)]]$ $(g(5)) =$

7. = $\lambda x[\neg\exists z[book(z) \wedge \lambda \mathbf{b}[like(z)(\mathbf{b})](x)]]$ $(g(5)) =$

8. = $\lambda \mathbf{x}[\neg\exists z[book(z) \wedge like(z)(x)]]$ $(\mathbf{g(5)}) =$

9. = $\neg\exists z[book(z) \wedge like(z)(g(5))]$

10. Let g(5) = the boys. Then

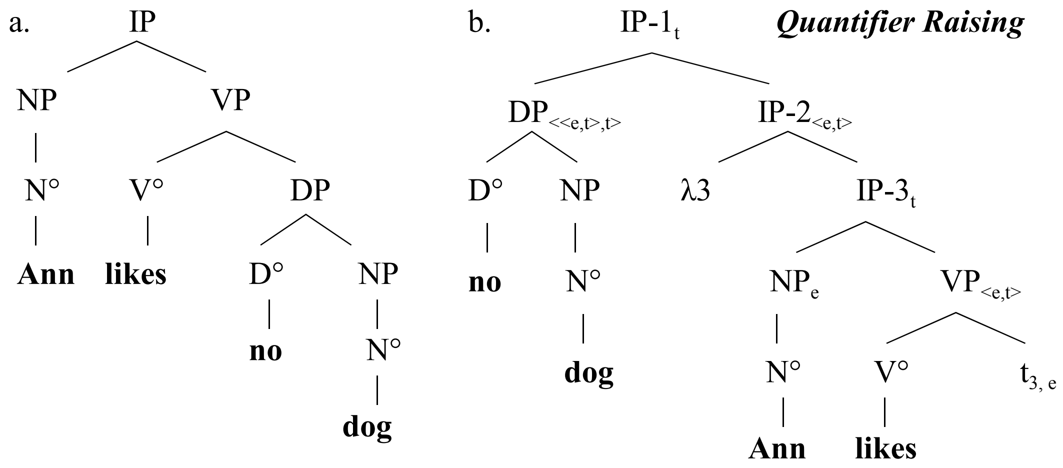
$\neg\exists z[book(z) \wedge like(z)(g(5))] = \neg\exists z[book(z) \wedge like(z)(\text{the boys})]$

Option II: Quantifier Raising

Option I requires a separate type shifting operation for QPs in direct object position, one for QPs in indirect object position, one for QPs which serve as prepositional complements, and so on. However, there is a second, more straightforward way to derive the meaning of sentences with QPs in object position, which makes use of the interpretational procedures which have already been introduced for the analysis of relative clause. In fact, it will turn out that - with the exception of a minor change in the definition of Predicate Abstraction - it is not necessary to add any further semantic assumption in order to arrive at the correct interpretation of object QPs!

To begin with, recall from the discussion of relative clause, that empty operator movement is interpreted as λ-abstraction over the trace left in the base position. Assume now that this method is generalized to other types of movement (wh-movement, topicalization, scrambling,...). Assume moreover that quantifiers move covertly in Logical Form (LF) and adjoin to the IP node (Chomsky 1976; May 1977, 1985; see Hornstein 1995 for overview). The LF-representation for sentence (4)b on this analysis is illustrated by the tree in (46)b (see Heim and Kratzer 1998: 184ff):

(46)



Before proceeding, a remark on the use of indices in the graph (46)b is in order: It is a common notational convention in the transformational grammars to subscript the index of a moved phrase to the trace as well as to the dislocated phrase. However, in (46)b, the movement index ('3') is separated from the moved category (*no dog*), and adjoined below the target of movement (*no dog*). This step is motivated by semantic considerations, as will become clear in a moment, and comes at the cost that movment does not just involve dislocation, but also implicates local reorganization of the tree. It should not go unnoticed, though, that the algorithm can be easily formalized and generalizes to all types of movement (e.g. operator movement).

Turning to the interpretation of (46)b, observe that after QR, the QP resides now in a position in which it is compositionally interpretable: Just as was the case with empty operator movement, the sister node of the dislocated category is a predicate of individuals (<e,t>) derived by λ-abstraction. The QP, which denotes a function of type <<e,t>,t>, can therefore directly combine with IP-2. So much for the types.

One more ingredient needs to be spelled out: the interpretation of λ-abstraction. The analysis of relative clause lead to the conclusion that an empty operator functions as λ-abstractor, and that variables bound by this λ-operator inside the relative clause are captured by the λ-operator by the means of a modified variable assignment ($[\![OP_n\ \Psi]\!]^g = \lambda x[[\![\Psi]\!]]^{g[n/x]}$). On current views, the rule for Predicate Abstraction needs to be slightly altered, leading to the - in fact simplified - version below:

> ***PA. Predicate (Lambda-) Abstraction*** ***(final version)***
>
> For any g and and $n \in \mathbb{N}$, if α has the form XP, then $[\![\alpha]\!]^g = \lambda x[[\![\gamma]\!]]^{g[n/x]}$
>
> n γ

Given these premises, the tree (46)b can now be subjected to semantic interpretation. Crucially, note that the interpretation of structures involving QR proceeds - up to IP-2 - exactly along the same lines as the interpretation of relative clause. The computation proceeds as follows (this proof is for one pretty complete, but still contains - for expository reasons - a redundant steps [2.]):

(47)
1. $[\![[_{IP\text{-}3}\ [\textbf{No dog}]\ [_{IP\text{-}2}\ \textbf{3}\ [_{IP\text{-}1}\ \textbf{Ann}\ [_{VP}\ \textbf{likes}\ \textbf{t}_3]]]]]\!]^g$ = 1 iff
2. $[\![[\textbf{No dog}]]\!]^g\ ([\![[_{IP\text{-}2}\ \textbf{3}\ [_{IP\text{-}1}\ \textbf{Ann}\ [_{VP}\ \textbf{likes}\ \textbf{t}_3]]]]\!]^g)$ = 1 iff 9. = 1 *FA*
3. $[\![[\textbf{No dog}]]\!]^g\ (\lambda x[[\![[_{IP\text{-}1}\ \textbf{Ann}\ [_{VP}\ \textbf{likes}\ \textbf{t}_3]]]\!]^{g[3/x]}])$ *λ-Abstraction*
4. $[\![\textbf{Ann}]\!]$ = Ann *Lexicon*
5. $[\![\textbf{likes}]\!]$ = $\lambda v\lambda w[like(v)(w)]$ *Lexicon*
6. $\lambda x[[\![[_{IP\text{-}1}\ \textbf{Ann}\ [_{VP}\ \textbf{likes}\ \textbf{t}_3]]]\!]^{g[3/x]}]$ =
   = $\lambda x[[\![\textbf{likes}]\!]\ ([\![\textbf{t}_3]\!]^{g[3/x]})(Ann)]$ = *Substitution & FA*
   = $\lambda x[[\![\textbf{likes}]\!]\ (g^{[3/x]}([\![\textbf{t}_3]\!]))(Ann)]$ = *Varibale assignment*
   = $\lambda x[[\![\textbf{likes}]\!]\ (g^{[3/x]}(3))(Ann)]$ = *Interpretation of trace*
   = $\lambda x[[\![\textbf{likes}]\!]\ (x)(Ann)]$ = *Application of* g
   = $\lambda x[\lambda v\lambda w[like(v)(w)](x)(Ann)]$ = *Substitution from 5.*
   = $\lambda x[\lambda w[like(x)(w)](Ann)]$ = *λ-Conversion*
   = $\lambda x[like(x)(Ann)]$ *λ-Conversion*
7. $[\![\textbf{dog}]\!]$ = $\lambda z[dog(z)]$ *Lexicon*
8. $[\![[\textbf{No dog}]]\!]^g = \lambda P\lambda Q\neg\exists y[P(y) \wedge Q(y)](\lambda z[\![dog(z)]\!])$ = *Lexicon & Substitution from 7.*
   = $\lambda Q\neg\exists y[\lambda v[\![dog(z)]\!](y) \wedge Q(y)]$ = *λ-Conversion*
   = $\lambda Q\neg\exists y[[\![dog(y)]\!] \wedge Q(y)]$ *λ-Conversion*
9. $\lambda Q\neg\exists y[dog(y) \wedge Q(y)]\ (\lambda x[likes(x)(Ann)])$ = *Substituting 6. & 8. into* 2.
   = $\neg\exists y[dog(y) \wedge \lambda x[likes\ (x)(Ann)](y)]$ =
   = $\neg\exists y[dog(y)] \wedge likes\ (y)(Ann)]]$

<u>Exercise</u>
The discussion above has revolved around cases in which one quantified phrase was situated either in subject or in object position of a transitive clause. But there are also cases which combine two or more QPs in one clause, as has been pointed out at various times. These sentences display truth -conditional ambiguity under certain circumstances. Determine for the following examples below:

(i) Whether they are ambiguous.
(ii) If they are, provide a compositional derivation (similar to the one given in (47)).
(iii) If they are not ambiguous, explain why.

(48)
a. Exactly five American presidents committed exactly one crime.
b. Some guard was standing in front of every window.
c. Every story has an ending.
d. Sally met a person who knew every member of the *Favoritener Kleintierzüchterverband*

10.4. MOTIVATING QR

Obviously, one would like to have independent evidence for the movement process of QP's postulated in 10.3. Otherwise, the analysis of object QPs in terms of QR would fail to satisfy the criterion that no assumption should be adopted just in order to justify the existence of the phenomenon under scrutiny. But there is in fact strong evidence in favor of such an operation which comes from the following constructions:

- QP-QP-interaction:

(49) A doctor saw each patient

- Antecedent Contained Deletion (May 197, 1985; Fiengo and May 1994)

(50) Sally met every person$_i$ Bill did ⌂
⌂ = met t_i

On ACD see below, section 11.

- Inverse Linking

(51) A barber from every city$_k$ hates it$_k$

- A constraint on QR: Coordinate Structure Constraint and Ruys (1992) effects

(52) a. Sam has seen none of the movies and Bill tried to see it
b. Sam has seen none of the movies and Bill tried to see Terminator.

Scenario: Assume that Sam saw Terminator and Scanners. Moreover, suppose Bill tried to see Scanners, but Bill didn't try to see Terminator (making the second conjunct come out as false). Finally, assume *it* refers to Terminator.

			Predicted	*Intuition*
(53)	a.	$\neg\exists x[\text{movie}(x) \wedge \text{see}(x)(\text{Sam}) \wedge \text{tried\_to\_see}(x)(\text{Bill})]$	F	✓
	b.	$\neg\exists x[\text{movie}(x) \wedge \text{see}(x)(\text{Sam}) \wedge \text{tried\_to\_see}(g(\text{it}))(\text{Bill})]$	T	✓
	c.	$\neg\exists x[\text{movie}(x) \wedge \text{see}(x)(\text{Sam})] \wedge \text{tried\_to\_see}(g(\text{it}))(\text{Bill})$	F	✓
(54)	a.	$\neg\exists x[\text{movie}(x) \wedge \text{see}(x)(\text{Sam}) \wedge \text{tried\_to\_see}(\text{Terminator})(\text{Bill})]$	T	*
	b.	$\neg\exists x[\text{movie}(x) \wedge \text{see}(x)(\text{Sam})] \wedge \text{tried\_to\_see}(\text{Terminator})(\text{Bill})$	F	✓

11. ANTECEDENT CONTAINED DELETION

11.1. PROBLEM & STANDARD ANALYSIS

Ellipsis resolution in configurations where the antecedent (the VP) contains the node which hosts the ellipsis (the object) leads to endless regress (May 1977, Fiengo & May 1994, Sag 1976):

(55) a. John $[_{VP}$ read every book Bill did ⌂]
b. ⌂ = $[_{VP}$ read every book Bill did ⌂]
c. John $[_{VP}$ read every book Bill did $[_{VP}$ read every book Bill did $[_{VP}$ read every book Bill did...]]] *Ellipsis resolution leads to endless regress*

Observation: ACD is possible only if the NP containing the ellipsis site - the *host* - is quantificational.

(56) a. Dulles suspected everyone that Philby does
b. Dulles suspected someone who Philby does
c. Dulles suspected noone who Philby does
d. *Dulles suspected Burgess who Philby does
e. *Dulles suspected his friend who Philby does
f. *Dulles suspected this spy who Philby does

This has been taken to indicate that ACD resolution should in one way or the other be tied to the process which fixes quantifier scope (QR).

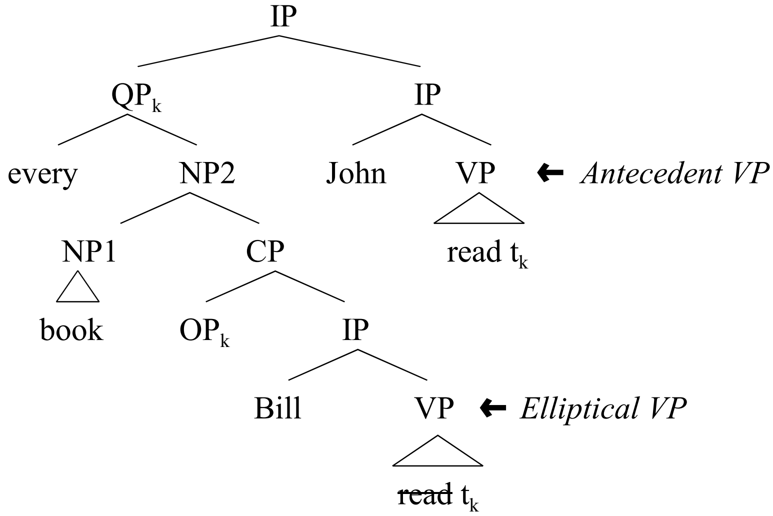
Standard Analysis: QR of QP which hosts the ellipsis site resolves regress (May 1977, Sag 1976):

(57) a. John $[_{VP}$ read every book that Bill did ⌂]
$[_{IP}$ [every book that Bill did ⌂$]_k$ $[_{IP}$ John $[_{VP}$ read t_k]] *QR*
b. ⌂ = $[_{VP}$ read t_k]
c. For every book such that Bill read it, John read it

● The fine-grained structure of (57)d look as in (59). Note that the constituent structure matches the interpretation given in (57)d (the empty operator OP_k in (59) represents the λ-abstractor). For reasons of clarity, the denotations of the components below are specified in set-notation.

(58) John $[_{\text{Antecedent VP}}$ read every book Bill $[_{\text{Elliptical VP}}$ did ⌂]]

(59)



(60) Meaning of relative clause: {x|Bill read x}

(61) Meaning of NP1 (book): {x|x is a book}

(62) Meaning of NP2 (book Bill read): {x|x is a book} ∩ {x|Bill read x} =
= {x|x is a book and Bill read x}

● Universal quantifier (*every*; formal symbol ∀) can be interpreted by subset relationship (‘⊆’):

(63) Every [dog] is [an animal]

{x|x is a dog} ⊆ {x|x is an animal}
“the set of dogs is a subset of the set of animals”

● Applying this algorithm to ACD yields the correct interpretation for the elliptical VP:

(64) a. John [$_{\text{Antecedent VP}}$ read every book Bill [$_{\text{Elliptical VP}}$ did ⌂]] *Surface representation*

b. every book Bill [$_{\text{Elliptical VP}}$ did ⌂] [John [$_{\text{Antecedent VP}}$ read t_k] *QR of object*

c. every book Bill [$_{\text{Elliptical VP}}$ did ⌂] [John [$_{\text{Antecedent VP}}$ read t_k]
[$_{\text{VP}}$ read t_k] *Fill antecedent VP into ellipsis site*

d. every book Bill [$_{\text{VP}}$ read t_k] [John [$_{\text{Antecedent VP}}$ read t_k]
{x|x is a book and Bill read x} ⊆ {x|John read x} *Interpretation*
“the set of books Bill read is a subset of the things John read”

11.2. Syntactic Locality

Observation I: The OP-chain in ACD is sensitive to locality conditions.

(65) a. *John [$_{\text{VP2}}$ read everything [$_{\text{CP}}$ which Bill believed [$_{\text{NP}}$ the claim that he [$_{\text{VP1}}$ did ⌂]]]]
b. [$_{\text{IP}}$ [everything [$_{\text{CP}}$ which Bill believed [$_{\text{NP}}$ the claim that he [$_{\text{VP1}}$ did ⌂]]]
[$_{\text{IP}}$ John [$_{\text{VP2}}$ read t_k] *QR*
c. ⌂ = [$_{\text{VP2}}$ read t_k]

This generalization supports the assumption that the ellipsis site contains a trace at LF, as well as structure to host this trace. Since QR is a precondition for identifying the ellipsis, the ellipsis site can furthermore be ‘filled’ by syntactic structure only subsequent to QR, i.e. at LF. Therefore, the locality violation in (65) cannot be computed earlier than at the abstract level of LF. Assuming that locality is a syntactic property, it follows that locality sensitivity of ACD can be interpreted as a further argument for the existence of an abstract syntactic level of representation (= LF).

Observation II: QR in ACD is sensitive to locality conditions.

● There are contexts in which QR can proceed either into a higher (infinitival) or into the lower clause. The ellipsis in ACD can then be resolved in two ways, leading to ambiguity:

(66) John expected the boys to read every book Mary [$_{VP}$ did ⌂]
  a. John expected the boys to read every book Mary read *Narrow QR*
  b. John expected the boys to read every book Mary expected the boys to read *Wide QR*

(67) John wanted to read every book Mary [$_{VP}$ did ⌂]
  a. John wanted to read every book Mary read *Narrow QR*
  b. John wanted to read every book Mary wanted to read *Wide QR*

● It is moreover possible to create contexts in which a narrow interpretation is blocked due to independent restrictions. Observe to begin with that the collective predicate *to agree on* in (68) is only compatible with a plural subject (**She/I/Tom/the man gathered*). More precisely, since there are two predicates subsequent to ellipsis resolution, both subjects must be plural:

(68) a. *She gathered in every city that Bill did
  b. *The girls gathered in every city that Bill did
  c. *She gathered in every city that the boys did
  d. The girls gathered in every city that the boys did

(69) combines properties of (66) with properties of (68): (69) is ambiguous w.r.t. ellipsis resolution, and at the same time contains a collective predicate. Since the subject of the relative clause in (69) is a singular term (*Bill*), the VP which fills the ellipsis site must be chosen such as to be compatible with a singular subject. Hence, only the higher VP, which does not require a plural subject, may serve as the antecedent for the ellipsis. Similar observations apply to (70) and (71)

(69) I expected the boys to gather in every city that Bill did
  a. *I expected the boys to gather in every city that *Bill agreed on*
  b. I expected the boys to gather in every city that
   Bill expected the boys to agree on

(70) I expected the boys to solve as a group every problem that Bill did *Not ambiguous*

(71) I expected the boys to agree on every topic Bill did *Not ambiguous*

● This configuration supplies (at least at first sight) a possibility to test the hypothesis that QR in ACD is sensitive to islands. More specifically, if the higher predicate and the QP are separated by an island, one is led to suspect that it should no longer be possible to generate the sensical reading, and the sentence should end up as unacceptable:

(72) *I had the expectation that the boys gather in every city that Bill did
  a. *I had the expectation that the boys gather in every city that
   Bill gathered in *Narrow QR*: ✘*Plural restriction*
  b. *I had the expectation that the boys gather in every city that
   Bill had the expectation that the boys gather in *Wide QR:* ✘*Locality*

The judgements about (72) appear rather robust. Unfortunately, the example can also be excluded by an additional factor: In (72), the empty operator chain also violates locality. Thus, one needs to find contexts which constitute islands for QR, but do not block empty operator movement.

● Does negation fulfill these criteria? Recall that negation blocks QR of objects:

(73) I do not know every poem. $\forall > \neg$ /*$\forall > \neg$

Consider in this light (74). On the one hand, the object needs to pass over negation in order to escape the plural predicate *solve as a group*. On the other hand, such movement should be blocked by negation. (74) indeed seems to be rather marked:

(74) ?*I don't expect the boys to gather in every city that Bill did

a. *I don't expect the boys to gather in every city that Bill gathered in *Narrow QR*: ✘*Plural restriction*

b. *I don't expect the boys to gather in every city that Bill didn't expect the boys to gather in *Wide QR:* ✘*Locality of QR*

Crucially, negation does not interfere with formation of relative clauses, thus the marginality of (74) cannot be due to a locality violation induced by empty operator movement:

(75) I know about a/every topic $\boldsymbol{OP_k}$ that Bill didn't solve $\boldsymbol{t_k}$

○ Similar conclusions can be reached on the basis of the following examples:

(76) ?*I didn't want them to agree on every topic Bill did

(77) ?*She claimed that the electorial committee elected every candidate Senator Smith did

In sum, these observations support the assumption that the host QP reaches its LF position by movement, and that this movement operation is subject to syntactic constraints. Next, consider a possible alternative analysis of ACD, which would - if successful - invalidate the argument for QR coming from ACD, and therefore deserves special attention.

11.3. EXTRAPOSITION

Alternative analysis: Baltin (1987) claims that ACD is an illusion. Ellipsis inside the relative clause is resolved in overt syntax by extraposition of the relative clause, which can equally aid in resolving endless regress:

(78) a. John [$_{VP}$ read every book that Bill did ⌂]
b. John [$_{VP}$ [$_{VP}$ read t_k] [every book that Bill did ⌂]$_k$] (Extraposition)
c. ⌂ = [$_{VP}$ read t_i]
d. For every book such that John read it, Bill read it

11.4. Larson & May (1990)

Larson and May present a number of counter-arguments against Baltin (1987), in defense of the original QR analysis. In sum, these (and other) considerations led to the widely shared consensus that ACD is in fact best analyzed as the product of QR.

Argument I: Why is extraposition obligatory? Usually, it is an optional process.

Argument II: Extraposed CPs do not license *that*-drop, while *that* may be omitted in the relative clause containing the ellipsis in ACD.

(79) I read something yesterday *(that) you had recommended

(80) I read something (that) John did

Argument III: Clause internal ACD cannot be reduced to extraposition, because there is direct evidence from serialization for the position of the relative clause - it is certainly not extraposed.

(81) a. ?John believed $[_{IP}$ [everyone you did ⌂] to be a genius]]
b. ⌂ = $[_{VP}$ you believed t to be a genius]

12. SOME LOGICAL PROPERTIES OF NL-QUANTIFIERS

(see e.g. von Fintel 1994)

There is an imbalance in the contribution which the left and the right argument of a quantifier make to the overall interpretation of the GQ. Various properties of GQ indicate that the left argument of the quantifier (its *domain* argument) plays a more prominent role.

12.1. RESTRICTIVE QUANTIFICATION

Natural language employs *restrictive quantification* (Mostowsky 1957, Rescher 1962, Geach 1972 among many others), as expressed by GQ theory:

(82) Most dogs sleep.
 a. Most individuals are such that they are dogs and sleep.
 b. Most individuals are such that if they are dogs then they sleep.

(83) *Question:* Which models would render (82)a and (82)b vacuously true?

(84) $[\![\text{most}]\!](A)(B) \Leftrightarrow |A \cap B| > \frac{1}{2}\, |A|$ [Fintel 1994]
 $[\![\text{most}]\!](A)(B) \Leftrightarrow |A \cap B| > |(D - B) \cap A|$ [Chierchia and McConnnell-Ginet 1990: 409]
 $[\![\text{most}]\!](A)(B) \Leftrightarrow |A \cap B| > |A - B\,|$ [Partee, ter Meulen, Wall 1993: 395]

12.2. CONSERVATIVITY

The left argument figures more prominently in interpretation:

(85) D is conservative iff ('live on' property of Barwise and Cooper 1981)
 $[\![D]\!](A)(B) \Leftrightarrow [\![D]\!]\,(A)(A \cap B)$

(86) *Conservative Quantifiers*
 a. ⟦Every man runs⟧ ⇔ ⟦Every man is a man who runs⟧
 b. ⟦No man runs⟧ ⇔ ⟦No man is a man who runs⟧
 c. ⟦Most men run⟧ ⇔ ⟦Most men are men who run⟧
 d. ⟦Fewer than 5 men run⟧ ⇔ ⟦Fewer than 5 men are men who run⟧

(87) *Non-conservative Quantifiers*
 a. $[\![\text{all non}]\!](A)(B) \Leftrightarrow (D\text{-}A) \subseteq B$ [Chierchia and McConnel-Ginet 1990]
 b. ¬⟦All non-men run⟧ ⇔ ⟦All non-men are men who run⟧

(88) *Question:* What is minimally required for the second sentence to be false?

(89) a. $[\![\text{only}]\!](A)(B) \Leftrightarrow B \subseteq A$
 b. ¬⟦Only men run⟧ ⇔
 c. ⟦Only men are men who run⟧ (2^{nd} clause is tautology)

12.3. Presuppositions

The denotation of the left argument of a quantifier can be restricted by presuppositions in that the set denoted by the left argument imposes necessary preconditions on the model:

(90) *Existence presupposition*
The king of France is bold

(91) *Cardinality presupposition*
a. If $|A| = 2$, then $[\![\text{both}]\!](A)(B) \Leftrightarrow A \subseteq B$
If $\neg\, |A| = 2$, then $[\![\text{both}]\!](A)(B)$ is undefined
b. Both dogs sleep.

Hence, formulas containing natural language quantifiers cannot be vacuously true (Strawson 1952):

(92) All dogs are asleep. (infelicitous in world without dogs)

12.4. A Consequence of Conservativity

Two further properties of a certain class of determiners:

(93) D is *symmetric* iff
$[\![D]\!](A)(B) \Leftrightarrow [\![D]\!](B)(A)$

(94) D is *intersective* iff
$[\![D]\!](A)(B) \Leftrightarrow [\![D]\!](A \cap B)(B)$

Symmetric D's are also intersective if conservativity is ensured (and v.v.):

(95) $[\![D]\!](A)(B) \Leftrightarrow [\![D]\!](B)(A) \Leftrightarrow [\![D]\!](A \cap B)(B)$

(96) symmetric → intersective
$[\![D]\!](A)(B) \underset{\text{symmetry}}{\Leftrightarrow} [\![D]\!](B)(A) \underset{\text{conservativity}}{\Leftrightarrow} [\![D]\!](B)(A \cap B) \underset{\text{symmetry}}{\Leftrightarrow} [\![D]\!](A \cap B)(B)$

(97) *Symmetric, intersective*
a. ⟦Some dogs sleep⟧ ⇔ ⟦Some sleepers are dogs⟧
b. ⟦Some dogs sleep⟧ ⇔ ⟦Some sleeping dogs sleep⟧

(98) *Non-symmetric, non-intersective*
a. ¬⟦All dogs sleep⟧ ⇔ ⟦All sleepers are dogs⟧
b. ¬⟦All dogs sleep⟧ ⇔ ⟦All sleeping dogs sleep⟧ (2nd clause is tautology)

(99) *Observation:* Symmetric, intersective D's are *weak*, they are tolerated in existential construction.

(100) There is/are some/a/three/*all/*every/*most dogs in the garden.

References

Baltin, Mark. 1987. Do Antecedent-Contained Deletions Exist. *Linguistic Inquiry* **18.**4: 579-596.

Barendregt, Hendrik Pieter. 1984. The Lambda Calculus. Amsterdam, NY: North-Holland.

Carnap, R. 1947. *Meaning and Necessity*. Chicago: Chicago University Press.

Chomsky, Noam. 1976. Conditions on Rules of Grammar. *Linguistic Analysis* **2**: 303-151.

Chomsky, Noam. 1986. *Knowledge of Language*. New York, New York: Praeger Publishers.

Fodor, Jerry. 1975. *The Language of Thought*. New York: Thomas Y. Crowell.

Church, Alonzo. 1940. A Formulation of a Simple Theory of Types. *Journal of Symbolic Logic* **5.** 56-68.

Dowty, David, Robert Wall and Stanley Peters. 1981. Introduction to Montague Semantics. Dordrecht: Reidel.

Euler, Leonhard. 1768. *Lettres a une Princesse d'Allemagne*. St. Petersburg.

Fiengo, Robert and Robert May. 1994. *Indices and Identity*. Cambridge, Massachusetts: MIT Press.

von Fintel, Kai. 1994. *Restrictions on Quantifier Domains*. Doctoral Dissertation: University of Massachusetts-Amherst.

Fox, Danny. 2002. Antecedent Contained Deletion and the Copy Theory of Movement. Linguistic Inquiry 33.1: 63-96.

Frege, Gottlob. 1892. Über Sinn und Bedeutung. *Zeitschrift für Philosophie und Philosophische Kritik* **100.** 25-50. also in *from the Philosophical Writings of Gottlob Frege,* pp. 56-78. Oxford: Basic Blackwell.

Frege, Gottlob. 1891. Function and Concept, *Translations from the Philosophical Writings of Gottlob Frege,* pp. 21-41. Oxford: Basic Blackwell.

Gamut, L.T.F. 1991. *Logic, Language and Meaning*. Chicago: University of Chicago Press.

Grayling, A.C. 1997. An Introduction to Philosophical Logic. Third Edition. Oxford: Blackwell

Heim, Irene and Angelika Kratzer. 1998. *Semantics in Generative Grammar*. Oxford: Blackwell.

Hornstein, Norbert. 1995. *Logical Form: From GB to Minimalism*. Cambridge, Massachusetts: Basil Blackwell.

Jackendoff, Ray. 1983. *Semantics and Cognition*. Cambridge, Massachusetts: MIT Press.

Jackendoff, Ray. 1996. The Proper Treatment of Measuring Out, Telicity, and Perhaps Even Quantification in English. *Natural Language and Linguistic Theory* **14.**2: 305-354.

Kamp, Hans and Uwe Reyle. 1993. From Discourse to Logic. Dordrecht: Kluwer Academic Publishers.

Larson, Richard and Robert May. 1990. Antecedent Containment or Vacuous Movement: Reply to Baltin. *Linguistic Inquiry* **21.**1: 103-122.

Lewis, David. 1975. Languages and Language. In K. Gunderson (Ed.), *Language, Mind and Knowlege,* pp. 3-35. Minneapolis: University of Minnesota Press.

Lycan, William G. 2000. Philosophy of Language. A Contemporary Introduction. London: Routledge.

May, Robert. 1977. The Grammar of Quantification. Doctoral Dissertation, Massachusetts Institute of Technology.

Montague, Richard. 1973. The Proper Treatment of Quantification in Ordinary English. In J. Hintikka, J. Moravcsik and P. Suppes (Eds.), *Approaches to Natural Language,* pp. 221-242. Dordrecht: Reidel.

Partee, Barbara and Mats Rooth. 1983. Generalized Conjunction and Type Ambiguity. In R. Baeuerle, C. Schwarze and A. con Stechow (Eds.), *Meaning and Interpretation of Language,* pp. 361-383. Berlin: de Gruyter.

Partee, Barbara, Alice ter Meulen and Robert Wall. 1990. *Mathematical Methods in Linguistics*. Dordrecht: Kluwer.

Russell, Bertrand. 1919. *Introduction to Mathematical Philosophy*. London: Allen and Unwin.

Sag, Ivan. 1976. *Deletion and Logical Form*. Doctoral Dissertation, Cambridge, Massachusetts: Massachusetts Institute of Technology.

Schönfinkel, Moses. 1924. Über die Bausteine der Mathematischen Logik. *Mathematische Annalen* **92.** 305-316.

Tarski, Alfred. 1935. Der Wahrheitsbegriff in den Formalisierten Sprachen, *Logic, Semantics, Metamathematics,* pp. 152-278. Oxford: Oxford University Press.

Taylor, Kenneth 1998. Truth and Meaning. An Introduction to the Philosophy of Language. Oxford: Blackwell.

Weis, William and Cherie D'Mello. 1997. *Fundamentals of Model Theory*: Ms., Department of Mathematics,University of Toronto. available at http://http://www.math.toronto.edu/weiss/

Handbooks & Overviews

Lappin, Shalom (ed.). 1996. The Interpretation of Ellipsis. *The Handbook of Contemporary Semantic Theory*. Oxford, Blackwell.

Partee, Barbara. 1999. "Semantics" in R.A. Wilson and F.C. Keil, eds., *The MIT Encyclopedia of the Cognitive Sciences*. Cambridge, MA: The MIT Press: 739-742.

von Stechow, A. and D. Wunderlich, Eds. (1991). *Semantik/Semantics. Ein internationales Handbuch der zeitgenoessischen Forschung/An International Handbook of Contemporary Research*. Berlin/New York, de Gruyter.

Some Web Resources (2005)

Standford Encyclopedia of Philosophy (maybe the best source for brief yet detailed introductions to various topics in the philosophy of language)
http://plato.stanford.edu/archives/fall2001/contents.html

Geschichte der Philosophie: http://www.philosophenlexikon.de/gesch.htm

Lexicon of Linguistics: http://tristram.let.uu.nl/UiL-OTS/Lexicon/

Philosophy Pages: http://www.philosophypages.com/index.htm

Semantics Archive: http://www.semanticsarchive.net/

Semanics Web Resources, collected by Kai von Fintel:
http://web.mit.edu/fintel/resources.html

von Stechow, Arnim. in progress. *Schritte zur Satzsemantik*:
http://www2.sfs.nphil.uni-tuebingen.de/arnim/

Zimmermann, Ede. Course material semantics:
http://www.uni-frankfurt.de/fb10/zimmermann/index.html

TABLE OF CONTENT